COMPUTE

the Club Of Microprocessor Programmers, Users, and Technical Experts

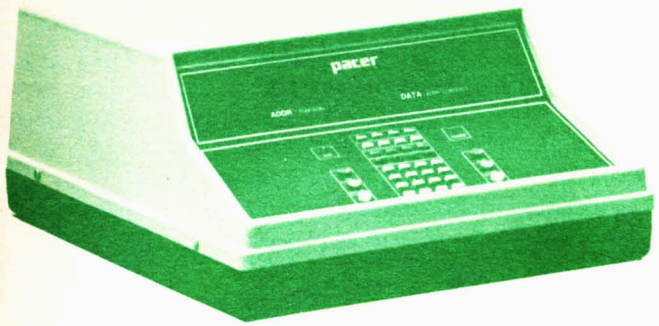
Georgia Marszalek, Editor • David Graves, Editor

Sponsored by National Semiconductor Corp., Santa Clara, Ca. 95051 **Vol. 2, No. 8, August, 1976**

16 BIT COMPUTER KIT EVALUATION

by John Snell

Reprinted with permission: People's Computer Company, P.O. Box 310, Menlo Park, CA 94025. July 17 Vol 5 No. 1.



I recently put together a Pacer 3H microcomputer development system kit, made by Project Support Engineering. It has by far the easiest to use operational and debug functions of any computer kit that I have used. It has a NOVUS type (no tactile feedback) calculator keyboard built into the front panel. When you want to deposit a number such as 7532, you hit the 7, 5, 3, 2 keys on the keyboard instead of binary switches 1,1,1, 1,0,1, 0,1,1, 0,1,0. The Pacer uses the hexadecimal number system for keyboard input and display. If you do not know hexadecimal, the keyboard will convert from decimal to hexadecimal as well as perform hexadecimal arithmetic. Addresses are displayed to the left and data contents to the right in alphanumeric LED's. Using the keyboard you may examine and/or modify not only memory locations but also:

- 4 accumulators X 16 bits,
- a 16 bit program counter,
- a 16 bit status register,
- a 10 word X 16 bit stack,
- a 16 bit value register used for scanning memory to find the address at which a value is located,
- a 16 bit mask (for use if you are not certain about some of the bits or digits in the value),
- and 10 break point registers, 16 bits each.

(Continued on page 10)

A HIGH LEVEL LANGUAGE FOR IMP-16

SM/PL (Smart or Simple Programming Language), a high level language for the IMP-16 Microprocessor, will be made available to COMPUTE members as part of the User Library. As such, it is not supported by National Semiconductor. SM/PL requires 16K of read/write memory, and can be used with the IMP-16 Disc Operating System. It accepts source statements generated by using EDIT16 and will product a listing of the assembly code generated or an object program. The compiler includes a number of features such as access to the IMP-16 machine facilities, linkage to assembly language routines and various compiler control statements for source listing, interlisting of assembly and object code, and symbol table dumps. A list of some of the language features is shown below.

SM/PL LANGUAGE FEATURES

Data Types:
BYTE – 8 bits
WORD – 16 bits, used for addresses or data

Arithmetic and Logical Operators:

+	Addition	NOT	Logical complementation
–	Subtraction	AND	Logical product
*	Multiplication	OR	Logical sum
/	Division	XOR	Logical exclusive or
MOD	Modulo		

Relational Operators:

<	Less than	<=	Less than or equal to
>	Greater than	>=	Greater than or equal to
=	Equal to	<>	Not equal to

Assignment Statements:

=	Simple assignment e.g., A = B	:=	Imbedded assignment e.g., A + (B:=+1)–C

Compound Statement:
A set of sequentially executed statements, treated as a unit, e.g., DO; A = 1, B = C + D*3; END;

IF-THEN-FALSE Statement:
Selects alternative paths of execution depending on a conditional test. Programmer defined flags, condition code settings, comparisons, logical operators, and arithmetic operations may be involved in the test. As with other control statements, IF-THEN-ELSE statements may be nested.

DO CASE Statement:
Selects a single statement to be executed from a list of statements, according to the value of a computed index.

(Continued on page 11)

Why you need "MICROBOARDS"

- *Complete* microprocessor system at minimal cost
- Pre-assembled and tested hardware
- 12V CMOS buffers on ALL Inputs AND Outputs providing excellent noise immunity
- Driver boards available to connect directly to motors, solenoids, lamps, etc., both AC and DC
- Software and hardware relationships designed for maximum utility
- New European standard board size

Why waste your capital developing your own microprocessor system when you can use the versatile 'MICROBOARD' system. Already proven in hundreds of hours trouble-free service, this system could get your new product into the market place way ahead of your competitors who are still struggling to learn the art of designing, building and debugging their own microprocessor systems.

Micropower Ltd. will give you all the assistance you need to help you get 'MICROBOARDS' working in your application. If you do not wish to write your own program we will discuss your requirements and supply sets of ready programmed PROMs to plug in for immediate use.

Specification (MPU Board):

Data Bus: 8-bit Tri-State data inputs. Active low.

Serial In: Single active low input to SC/MP SIN pin.

Multiplexed: Input 1 SP C/O (microswitch) input or direct. (See note.)
Input 2 SP C/O (microswitch) input or direct.
Input 3 SP C/O (microswitch) input or direct.
Input 4 SP C/O (microswitch) input or direct.
Input 5 Active low.
Input 6 Active low.
Inputs 2 through 5 are all interrupt inputs.

Note: Any of inputs 1 through 4 can be either a single active low input or a single pole changeover by simply changing a wire link on the board.

OUTPUTS: *All* 12V CMOS levels.

Data Bus: 8-bit active high outputs. CMOS buffered directly from system data bus.

Serial Out: Single active high output from SC/MP SOUT pin.

Decoder: 16 alternate directly addressable latched outputs. By the simple changing of a wire link on the board, these 16 outputs become strobe outputs for the 8-bit latches on the optional driver boards. Compatible driver boards provide up to 120 parallel outputs with master reset facility.

Memory: 256 bytes (8-bits) Random Access Memory. 4 PROM sockets for MM5204Q programmable UV erasable 512 X 8-bit memories. Providing a total program capacity of 2048 bytes.

Interrupt: Four levels of interrupt are directly accessible as inputs without the need for additional decoding or hardware interrogation circuitry.

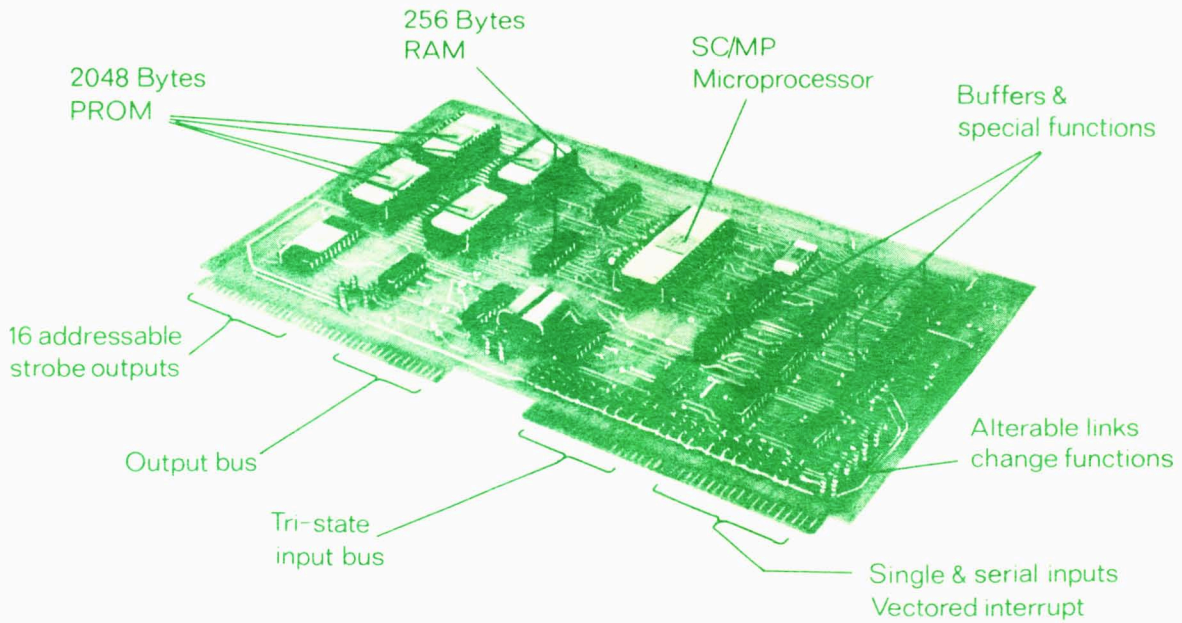
Additional Features: Power-up or manual reset-single cycle facility. Inputs expandable.

Power Supply Requirements: +5V 150mA
−7V 60mA
+12V 20mA
−12V 100mA

MICROBOARDS—with SC/MP MICROPROCESSOR

The versatile, intelligent control system

Let 'MICROBOARDS' with SC/MP work for you.



Physical Characteristics: Epoxy glass Printed Circuit Board, size Eurocard 'large' 233.4mm X 160mm.
Two single sided 40-way gold plated edge connector pads on 2.54mm (0.1") pitch.
Boards supplied with all components fully assembled and tested, including SC/MP, but less PROMs.

Obtainable from:
N SIGN COMPONENTS LTD.
P.O. Box 119, Reading, Berks. RG3 1NQ. Great Britain
Tel: Reading 594911/2/3/4 Telex: 849391

MICROPOWER LIMITED,
P.O. Box 39, Reading, Berks. RG3 6QF.Great Britain
Tel: Reading 21437

NEW LIBRARY PROGRAM FOR PACE, SL0025A PALM

The SL0025A PALM is a PACE program that can be used for dumping memory location onto paper tape in absolute format. The paper tape can be loaded by ABSPT. This program will also dump itself. Source (PT): $5.00

PALM EXECUTION EXAMPLE
TYPE IN THE RANGE 0000:03FF
ENTRY 0000 (Turn Punch On)

```
END PASS 1
    1                           .TITLE  PALM,'ABSOLUTE LM PUNCH'
    2       0000                .TSECT
    3       0000                R0=0
    4       0001                R1=1
    5       0002                R2=2
    6       0003                R3=3
    7 0000 957C A  PALM:  JSR   @MESG
    8 0001 0069 T         .WORD RANGE
    9 0002 155A A         JSR   GET
   10 0003 D575 A         ST    R1,POS
   11 0004 9577 A         JSR   @GECO
   12 0005 1557 A         JSR   GET
   13 0006 D56C A         ST    R1,END
   14 0007 9575 A         JSR   @MESG
   15 0008 007F T         .WORD STAR
   16 0009 1553 A         JSR   GET
   17 000A D573 A         ST    R1,START
   18 000B 5000 A         LI    R0,0
   19 000C D16B A         ST    R0,CHK
   20 000D 5132 A         LI    R1,50
   21 000E 153E A         JSR   NULL
   22 000F 5002 A         LI    R0,2
   23 0010 9569 A         JSR   @PUTC
   24 0011 5000 A         LI    R0,0
   25 0012 9567 A         JSR   @PUTC
   26 0013 5002 A         LI    R0,2
   27 0014 9565 A         JSR   @PUTC
   28 0015 5108 A         LI    R1,8
   29 0016 1536 A         JSR   NULL
   30 0017 1530 A         JSR   EOR
   31 0018 5002 A  DATAS:  LI    R0,2
   32 0019 9560 A         JSR   @PUTC
   33 001A 5080 A         LI    R0,080
   34 001B 955E A         JSR   @PUTC
   35 001C 5010 A         LI    R0,010
   36 001D 955C A         JSR   @PUTC
   37 001E 1533 A         JSR   CSUM
   38 001F 5102 A         LI    R1,2
   39 0020 152C A         JSR   NULL
   40 0021 C157 A         LD    R0,POS
   41 0022 9558 A         JSR   @PUT2C
   42 0023 5104 A         LI    R1,4
   43 0024 1528 A         JSR   NULL
   44 0025 530C A         LI    R3,12
   45 0026 C952 A         LD    R2,POS
   46 0027 C280 A  DATA:   LD    R0,(R2)
   47 0028 9552 A         JSR   @PUT2C
   48 0029 7A81 A         AISZ  R2,1
   49 002A 7BFF A         AISZ  R3,-1
   50 002B 19FB A         JMP   DATA
   51 002C 151B A         JSR   EOR
   52 002D D94B A         ST    R2,POS
   53 002E C14A A         LD    R0,POS
   54 002F 9D43 A         SKG   R0,END
   55 0030 19E7 A         JMP   DATAS
   56 0031 5000 A         LI    R0,0
   57 0032 D146 A         ST    R0,POS
   58 0033 5002 A         LI    R0,2      ;END RECORD
   59 0034 9545 A         JSR   @PUTC
   60 0035 50C0 A         LI    R0,0C0
   61 0036 9543 A         JSR   @PUTC
   62 0037 5004 A         LI    R0,4
   63 0038 9541 A         JSR   @PUTC
   64 0039 C13E A         LD    R0,CHK
   65 003A E143 A         ADD   R0,START
   66 003B 953F A         JSR   @PUT2C
   67 003C 5102 A         LI    R1,2
   68 003D 150F A         JSR   NULL
   69 003E C13F A         LD    R0,START
   70 003F 953B A         JSR   @PUT2C
   71 0040 5102 A         LI    R1,2
   72 0041 1508 A         JSR   NULL
   73 0042 C135 A         LD    R0,CHK
   74 0043 9537 A         JSR   @PUT2C
   75 0044 1503 A         JSR   EOR
   76 0045 5132 A         LI    R1,50
   77 0046 1506 A         JSR   NULL
   78 0047 0000 A         HALT
   79 0048 C12E A  EOR:    LD    R0,CR
   80 0049 9531 A         JSR   @PUT2C
   81 004A 5108 A         LI    R1,8
   82 004B 1501 A         JSR   NULL
   83 004C 8000 A         RTS
   84 004D 5000 A  NULL:   LI    R0,0
   85 004E 952B A         JSR   @PUTC
   86 004F 79FF A         AISZ  R1,-1
   87 0050 19FC A         JMP   NULL
   88 0051 8000 A         RTS
   89 0052 C926 A  CSUM:   LD    R2,POS
   90 0053 53F4 A         LI    R3,-12
   91 0054 C124 A         LD    R0,POS
   92 0055 E200 A         ADD   R0,(R2)
   93 0056 7A81 A         AISZ  R2,1
   94 0057 7B01 A         AISZ  R3,1
   95 0058 19FC A         JMP   CSUM+3
   96 0059 9521 A         JSR   @PUT2C
   97 005A E11D A         ADD   R0,CHK
   98 005B D11C A         ST    R0,CHK
   99 005C 8000 A         RTS
  100 005D 5304 A  GET:    LI    R3,4
  101 005E 951D A         JSR   @GECO
  102 005F A915 A         AND   R0,MASK1
  103 0060 9D15 A         SKG   R0,K39
  104 0061 1901 A         JMP   .+2
  105 0062 7809 A         AISZ  R0,9
  106 0063 A910 A         AND   R0,MASK
  107 0064 2908 A         SHL   R1,4,0
  108 0065 5900 A         RXOR  R0,R1
  109 0066 7BFF A         AISZ  R3,-1
  110 0067 19F6 A         JMP   GET+1
  111 0068 8000 A         RTS
  112 0069 5459 A  RANGE:  .ASCII  'TYPE IN THE RANGE '
      006A 5045 A
      006B 2049 A
      006C 4E20 A
      006D 5448 A
      006E 4520 A
      006F 5241 A
      0070 4E47 A
      0071 4520 A
  113 0072 0000 A         .WORD 0
  114 0073 0000 A  END:    .WORD 0
  115 0074 000F A  MASK:   .WORD 0F
  116 0075 007F A  MASK1:  .WORD 07F
  117 0076 0039 A  K39:    .WORD 039
  118 0077 0D0A A  CR:     .WORD 0D0A
  119 0078 0000 A  CHK:    .WORD 0
  120 0079 0000 A  POS:    .WORD 0
  121 007A 7E44 A  PUTC:   .WORD 07E44
  122 007B 7EC1 A  PUT2C:  .WORD 07EC1
  123 007C 7E59 A  GECO:   .WORD 07E59
  124 007D 7EA7 A  MESG:   .WORD 07EA7
  125 007E 0000 A  START:  .WORD 0
  126 007F 454E A  STAR:   .ASCII  'ENTRY '
      0080 5452 A
      0081 5920 A
  127 0082 0000 A         .WORD 0
  128      0000           .END  PALM

CHK     0078 T    CR      0077 T    CSUM    0052 T
DATA    0027 T    DATAS   0018 T    END     0073 T
EOR     0048 T    GECO    007C T    GET     005D T
K39     0076 T    MASK    0074 T    MASK1   0075 T
MESG    007D T    NULL    004D T    PALM    0000 T
POS     0079 T    PUT2C   007B T    PUTC    007A T
R0      0000 A    R1      0001 A    R2      0002 A
R3      0003 A    RANGE   0069 T    STAR    007F T
START   007E T

NO ERROR LINES
END PASS 4
SOURCE CHECKSUM=F0FA
OBJECT CHECKSUM=0248
```

Conversational Assembler

by Ed Schoell
National Semiconductor, Australia

This program can be used with the IMP-16 editor, PACE editor, IMP-16 CASM or PACE CASM.

TABTAPE overlays the card reader commands of EDIT 16 with the TT command. This command punches a paper tape and replaces tabs (control 1) with one tab character. This feature saves tape and assembly time by eliminating the tab character. It acts as a replacement for the PT command used with EDIT 16. Source: $5.00

```
NSC CONVERSATIONAL ASSEMBLER 1/2/75
Sample output when TABTAB is used with
the conversational assembler
   ?  KB
->          .TITLE  TEST
->  START:  LI      0,5
->          AISZ    0,1
->          JMP     START
->          .END    START
->    ***
```

Tape Program

```
   ?  LS
   1          .TITLE  TEST
   2   START: LI      0,5
   3          AISZ    0,1
   4          JMP     START
   5          .END    START
```

List Edit Buffer

```
   ?  TL
TURN PUNCH ON

   ?  TT
TURN PUNCH ON
.TITLETEST
START:LI0,5
AISZ0,1
JMPSTART
.ENDSTART

   ?  TL
TURN PUNCH ON
```

Punch Tape without Tab Characters

```
   ?  CR
```

Clear Buffer

```
   ?  RT
```

Read Tape Punched with TT Command

```
   ?  LS
   1          .TITLE  TEST
   2   START: LI      0,5
   3          AISZ    0,1
   4          JMP     START
   5          .END    START

   ?
```

List Tape with Tab Characters

PACE, IMP-16 ASSEMBLER SL0026A

```
END PASS 1
    1                   .TITLE  TABTAP,' VERS A   18 DEC 75'
    2              ;
    3              ;
    4              ;*************************************************
    5              ;
    6              ;
    7              ;       THIS PROGRAM REDUCES THE LENGHT OF THE SOURCE
    8              ;       TAPES FROM THE IMP & PACE EDITORS BY USING A
    9              ;       SINGLE TAB CHARACTER TO REPLACE A STRING
   10              ;       OF SPACES. THIS PROGRAM OVERWRITES THE CARD
   11              ;       READER ROUTINES IN THE EDITORS AND MAY BE
   12              ;       LOADED BY PRESSING 'LOAD PROG' AGAIN AFTER TH
   13              ;       EDITOR IS LOADED.
   14              ;
   15              ;
   16              ;*************************************************
   17              ;
   18              ;
   19              ;       IMP 16L/P  EDIT 16 **IMPEDT**
   20              ;
   21              ;       IMP 16L/P  CASM    **IMPCSM**
   22              ;
   23              ;       PACE 16P   EDIT    **PACEDT**
   24              ;
   25              ;       PACE 16P   CASM    **PCASM **
   26              ;
   27              ;
   28      0000            .ASECT
   29              ;
   30              ;
   31              ;       PROGRAM SELECTION DEFINITIONS
   32              ;
   33      0000    IMPEDT  =       0       ;TYPE 1 FOR IMP16 EDITOR
   34      0001    IMPCSM  =       1       ;TYPE 1 FOR IMP16 CONV EDT/ASM
   35      0000    PACEDT  =       0       ;TYPE 1 FOR PACE EDITOR
   36      0000    PCASM   =       0       ;TYPE 1 FOR PACE CONV EDT/ASM
   37              ;
   38              ;
   39              ;
   40              ;                       DEFINITIONS
   41              ;
   42      0000     000D                   .IF     IMPEDT
   43
   44
   45
   46
   47
   48
   49                              .ENDIF
   50              ;
   51              ;
   52      0001     000D                   .IF     IMPCSM
   53      00B1    DEVICE  =       0B1
   54      13BA    PUNCHL  =       013BA
   55      0FB0    START   =       0FB0
   56      1387    INERR   =       01387
   57      101C    KBMODE  =       0101C
   58      00D2    SPUTC   =       0D2
   59                              .ENDIF
   60              ;
   61              ;
   62      0000     000D                   .IF     PACEDT
   63
   64
   65
   66
   67
   68
   69                              .ENDIF
   70              ;
   71              ;
   72      0000     000D                   .IF     PCASM
   73
   74
   75
   76
   77
   78
   79                              .ENDIF
   80              ;
   81              ;
   82              ;
   83
                                   .PAGE
   84              ;
   85      102B                    .=      KBMODE + 0F
   86 102B 5454 A                  .ASCII  'TT'
   87 102C 1387 A                  .WORD   TABTAPE
   88      1387                    .=      INERR
   89 1387 892B A  TABTAPE:LD      2,ATOUT
   90 1388 A8B1 A                  ST      2,DEVICE
   91 1389 4C00 A                  LI      0,0
   92 138A A127 A                  ST      0,FLAG
   93 138B A123 A                  ST      0,SPCT
   94 138C 212D A                  JMP     PUNCHL
   95 138D F123 A  TOUT:   SKNE    0,SPACE
   96 138E 210A A                  JMP     STSPACE
   97 138F 7D22 A                  DSZ     FLAG            ;ARE SPACES ACCUMULAT
   98 1390 2106 A                  JMP     PUTCH
   99 1391 A11E A                  ST      0,CHAR          ;SAVE CHARACTER WHILE
100                                                        ;SENDING SPACES
101 1392 811E A                  LD      0,SPACE
102 1393 2CD2 A  SPCONT: JSR     @SPUTC          ;SEND ACCUMULATED
103                                              ;SPACES
104 1394 7D1A A                  DSZ     SPCT
105 1395 21FD A                  JMP     SPCONT
106 1396 811A A                  LD      0,CHAR          ;SEND CHAR AFTER SPAC
107 1397 2CD2 A  PUTCH:  JSR     @SPUTC
108 1398 0200 A                  RTS
109 1399 A11A A  STSPACE:ST      0,SAV0          ;SAVE CHAR IN AC0
110 139A F911 A                  SKNE    2,TAB1
111 139B 2109 A                  JMP     TAB
112 139C F910 A                  SKNE    2,TAB2
113 139D 2107 A                  JMP     TAB
114 139E F90F A                  SKNE    2,TAB3
115 139F 2105 A                  JMP     TAB
116 13A0 4C01 A                  LI      0,1
117 13A1 A110 A                  ST      0,FLAG          ;SET SPACE FLAG
118 13A2 790C A                  ISZ     SPCT
119 13A3 8110 A                  LD      0,SAV0          ;RESTORE AC0
120 13A4 0200 A                  RTS
121 13A5 4C09 A  TAB:    LI      0,09
122 13A6 2CD2 A                  JSR     @SPUTC          ;SEND A 'TAB' CHAR
123 13A7 4C00 A                  LI      0,0
124 13A8 A106 A                  ST      0,SPCT          ;RESET SPACE COUNTER
125 13A9 A108 A                  ST      0,FLAG          ;RESET SPACE FLAG
126 13AA 8109 A                  LD      0,SAV0          ;RESTORE AC0
127 13AB 0200 A                  RTS
128 13AC FFF9 A  TAB1;   .WORD   -7
129 13AD FFF1 A  TAB2;   .WORD   -15
130 13AE FFE1 A  TAB3;   .WORD   -31
131 13AF 0000 A  SPCT;   .WORD   0
132 13B0 0000 A  CHAR:   .WORD   0
133 13B1 0020 A  SPACE:  .WORD   020
134 13B2 0000 A  FLAG:   .WORD   0
135 13B3 138D A  ATOUT:  .WORD   TOUT
136 13B4 0000 A  SAV0:   .WORD   0
137      0FB0            .END    START

ATOUT   13B3  A         CHAR    13B0  A         DEVICE  00B1  A
FLAG    13B2  A         IMPCSM  0001  A         IMPEDT  0000  A
INERR   1387  A         KBMODE  101C  A         PACEDT  0000  A
PCASM   0000  A         PUNCHL  13BA  A         PUTCH   1397  A
SAV0    13B4  A         SPACE   13B1  A         SPCONT  1393  A
SPCT    13AF  A         SPUTC   00D2  A         START   0FB0  A
STSPAC  1399  A         TAB     13A5  A         TAB1    13AC  A
TAB2    13AD  A         TAB3    13AE  A         TABTAP  1387  A
TOUT    138D  A

NO ERROR LINES
END PASS 2
SOURCE CHECKSUM=6DFF
OBJECT CHECKSUM=6C16
```

Which course should you attend?

Three types of courses are offered by National: Fundamentals, Applications, and Programming. Each course addresses different training objectives, so you should carefully select the course or courses which meet your particular objectives. Our courses are described briefly below, and are described in detail in the *Course Specifications* section of this catalog.

Microprocessor Fundamentals
This course is designed for the engineer, senior technician, or manager who has no previous experience with program alterable systems. This is the course to attend for

- An introduction to microprocessor-based systems design
- An introduction to microprocessor programming, including hands-on experience with development systems and familiarization with support software packages (assemblers, editors, loaders, debugs).
- Guidelines for selecting a microprocessor.

SC/MP Applications
If you want to learn how to design systems using the SC/MP microprocessor, or if you want to thoroughly evaluate SC/MP for your applications, you should attend this course. In this course you will learn

- Hardware design techniques using SC/MP.
- SC/MP programming, including the use of development systems and support hardware.
- How to build SC/MP applications systems, since you will design, build, program, and debug applications systems.

PACE/IMP-16 Applications
In this course you will learn how to design systems using the PACE and IMP-16 microprocessors. Since PACE and IMP-16 are used in systems that require powerful computational capability, class problems are designed to show you how to use the sophisticated features of these microprocessors. Attend this course if you want to

- Learn to design systems using PACE or IMP-16.
- Evaluate PACE and IMP-16 for your applications.

Advanced Programming
This course is designed for the engineer who has mastered the hardware portions of microprocessor-based systems, and has a good foundation in software. It is also for the programmer who has been exposed only to batch-mode programming, or who needs more experience in the direct control of I/O devices. This course is effectively an extension of the PACE/IMP-16 APPLICATIONS course and the SC/MP APPLICATIONS course. Attend this course for

- Real-time programming techniques
- I/O control techniques, including interrupt handling.
- The use of more powerful development tools for applications software.

In-House Courses
Any of the courses that are offered at our training centers can be conducted at your facility, or we can create a special course to meet your specific needs. If you have several people who must be trained, it will probably be more economical for us to bring the training center to you.

When you contract with us to conduct a course at your facility, we provide

- Lab stations equivalent to those used in our permanent training centers.
- The same highly qualified instructors who conduct our resident courses.
- A training session tailored to meet your specific objectives.

You provide

- Lecture and lab rooms with appropriate furniture and training aids.
- Some eager students.

Costs for in-house courses vary according to the duration of the course, the number of students, and the amount of lab equipment which we must provide, so each course must be quoted individually. Contact your local National Sales Representative, and he can have a quote prepared for you.

Western Training Center

Location
National's Western Training Center is located in Santa Clara, California, forty miles south of San Francisco and five miles north of San Jose. The address is

Western Training Center, MS 470
National Semiconductor Corporation
2900 Semiconductor Drive
Santa Clara, California 95051
Telephone: (408) 737-5889

Eastern Training Center

Location
National's Eastern Training Center is located in Coral Gables, Florida, near Miami. The address is

Eastern Training Center
National Semiconductor Corporation
1320 South Dixie Hwy., Suite 870
Coral Gables, Florida 33146
Telephone: (305) 661-7969 or 661-7971

SCHEDULE OF MICROPROCESSOR CLASSES

	Eastern Training Center	Western Training Center
Microprocessor Fundamentals	September 27-October 1 November 8-12	September 13-17 October 25-29
SC/MP Applications	October 4-8 November 15-19	September 20-24 November 8-12
PACE Applications	October 18-22	September 27-October 1 November 1-5
Advanced Programming	August 23-27	October 3-8

New Product Information

AEG Telefunken, Department of Avionics and Special Function, announces the introduction of a new and inexpensive 8-bit microprocessor system on Euro-Cards (100 by 160 mm) in a 19-inch rack.

At the present time, the system has the following features:

1 • Microprocessor card based on National Semiconductor's SC/MP, containing:
  - Bus driver logic
  - Reset logic
  - Preaddressing selection (64 Addr. Max.)
  - 1/4k RAM
  - 1k ROM
  - TTY-Interface
  - DMA multiprocessing is possible

2 • ROM memory card – 8k ROM + 1/4k RAM (max. development 32k PROM)

3 • RAM memory card – 4k RAM (max. development: 16k RAM)

4 • TTL I/O card – 16 bit each
  - Frame to take up to 10 component groups + power supply

Preliminary Price List

8-bit Microprocessor Group (ROM not included)	$ 352 FOB Germany
4k RAM Group	$ 368 FOB Germany
ROM Group (ROM not included)	$ 216 FOB Germany
I/O Group	$ 196 FOB Germany
Frame + Power Supply	$1060 FOB Germany

Availability, September 1976

Contact:
Mr. Beis
AEG Telefunken
Abt. V225
Industiestrasse 29
2000 Wedel
Hollstein, Germany

SC/MP Kit Replacement Parts and Repair Policy

1. No distributor returns of assembled kits will be accepted. An open skinpac constitutes an assembled kit.
2. Replacement parts are available from the microprocessor service center in the following configuration. All other parts are available through local distributors.

   A. PC Board P/N 5514879 @ $25
   B. Programmed ROM P/N 4100937 @ $25
3. Assembled units may be returned for repair at a $25 per unit charge.

   All parts requests and units to be repaired should be sent to:

   Microprocessor Service Center
   2921 Copper Road
   Santa Clara, CA 95051
   Attention: Service Manager
4. A check for the total amount made out to National Semiconductor must accompany each parts request or unit to be repaired.

Memory Addressing In PACE Microprocessor

Part of the PACE microprocessor's powerful instruction set is a flexible method of addressing the memory. This method makes it possible to reference three sequences of 256 words located anywhere in the 65,536-word memory, as well as another 256 words in fixed positions.

The fixed words from what is called a "base" page, and the others form three "floating" pages. The mode of addressing is specified by the 2-bit XR field (bits 8 and 9) of the 16-bit instruction, as shown in the figure.

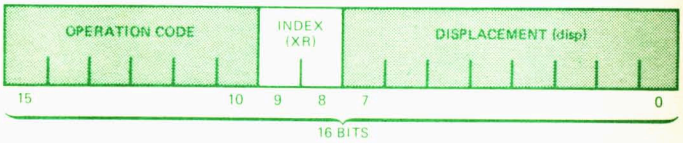
When the XR field is 00, it specifies base-page addressing. The base page may consist of either the first 256 words in the memory, or the first 128 plus last 128 words. The base-page-select (BPS) signal input decides which option will be used.

To address the first 256 words of memory (locations 0-255), BPS is set to 0, and the 16-bit memory address is formed by setting bits 8 through 15 to zero and by using bits 0 through 7 to specify one of 256 locations.

If BPS is 1, the 16-bit memory address is formed by setting bits 8 through 15 equal to bit 7 and by using 0 through 6 to locate the first 128 words of the memory (when bit 7 is 0) and the last 128 words (when bit 7 is 1). This technique is useful for splitting the base page between read-write and read-only memories or between memory and peripheral devices, so convenient base-page addressing can access data or peripherals.

When the XR field is 01, it specifies that addressing be relative to the program counter (PC). In this mode, the memory address is formed by adding the contents of the program counter to the value of bits 0 through 7, treated as a two's complement number with sign. That is, the bits 0 to 7 are interpreted as a 16-bit value with bits 8 through 15 set equal to bit 7. This allows numbers from –128 through +127 to be represented. Bits 0 to 7 are called displacement bits, since they can represent a range of words around a center position.

When the XR field is 10 or 11, addressing is relative to an index register, and any memory location within the external 65,536-word address space may be referenced. As before, the displacement field is interpreted as a signed value ranging from –128 through +127. The memory address is then formed by adding the displacement bits to the contents of either accumulator AC 2 (when XR = 10) or accumulator AC3 (when XR = 11). This type of addressing is desirable for those applications that require addresses to be computed at execution time, since addresses can not be modified when a ROM is serving for program storage (as is usually the case with microprocessors as opposed to minicomputers).



XR FIELD	ADDRESSING MODE	EFFECTIVE ADDRESS
00	Base page	EA = disp
01	Program-counter relative	EA = disp + (PC)
10	AC2 relative (indexed)	EA = disp + (AC2)
11	AC3 relative (indexed)	EA = disp + (AC3)

PRELIMINARY DATA: JULY 1976
Keith Winter and Milt Schwartz
National Semiconductor Corp.

SC/MP Mates with Cassette Recorder

introduction

This application note presents an inexpensive, reliable, and flexible method of storage of digital data and computer programs on cassette tapes as an alternative to using paper tape, PROMs, or other more-complex and more-expensive media. As an example of one benefit of using this method, information may be easily stored and transported, thereby making it easy to lend or trade programs.

To demonstrate this application, the SC/MP Low Cost Development System (LCDS) was used. All interface circuits, decoding, and RAMs or PROMS were breadboarded on a wirewrapped prototyping card. (See figure 8 for photographs of breadboard and table 2 for parts list.) This wirewrapped card is compatible with the 72-pin connectors that are contained in the SC/MP LCDS. Off-the-shelf integrated circuits were used to implement the circuits on the breadboard.

A $50 cassette recorder provides satisfactory performance. However, the consistent performance of a recorder in the $80 to $100 range may be more desirable. (See table 1 for a list of recorders used in verifying this application.) A 30-minute cassette allows approximately 40K 8-bit bytes of storage per side of cassette tape. This means that 17 2K-byte programs (including inter-program gaps) may be stored per side of tape. Reliability of transmission at a 330-baud rate (40 bytes/second) should be sufficient for most hobbyists and for some engineering development systems. To test the reliability, a 10K-byte program was loaded 10 times in succession. For this test, four cassette recorders were used; one recording and two playbacks per recorder were made. For each recording, the program was played back on a different cassette than was used for recording. (See table 1.) The results were error-free in all cases.

Figure 1, SC/MP-to-Cassette System Block Diagram, shows the major units for this application and their interrelations. Figures 2 through 4 are detailed schematic diagrams of the breadboard portion of the system. The interconnection for this application is shown in figure 1.

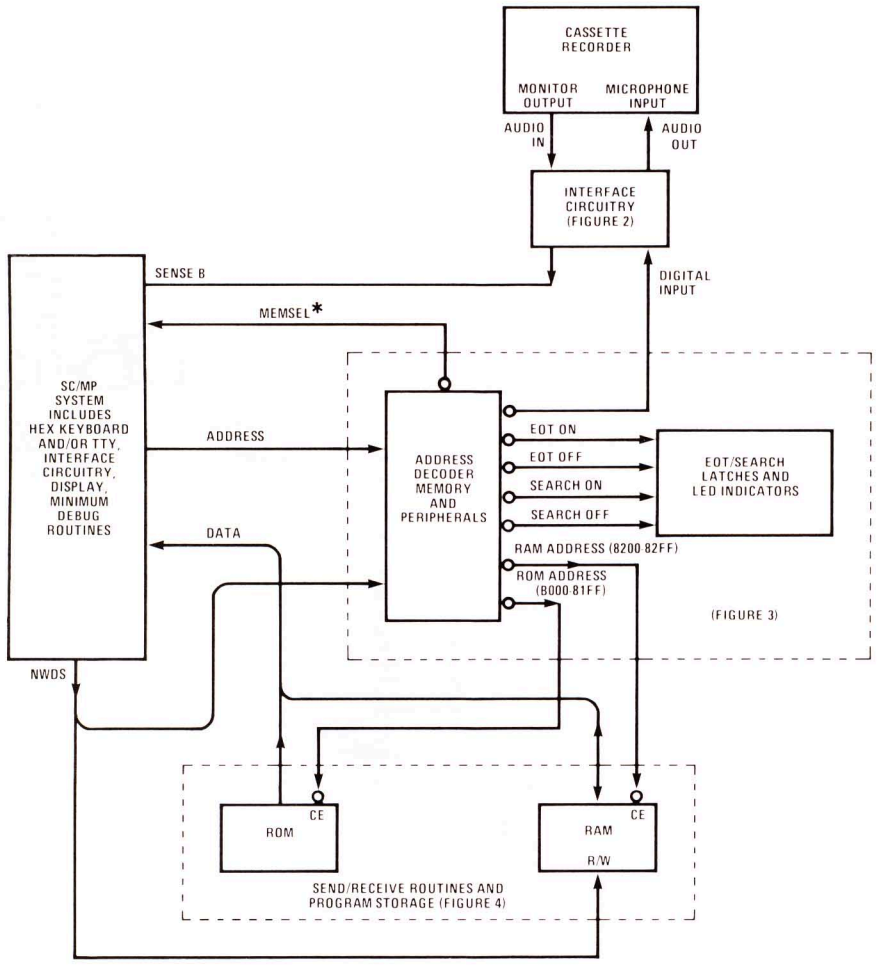


Figure 1. SC/MP-to-Cassette System Block Diagram

© 1976 National Semiconductor Corp.
DA-B20M86/Printed in U.S.A.

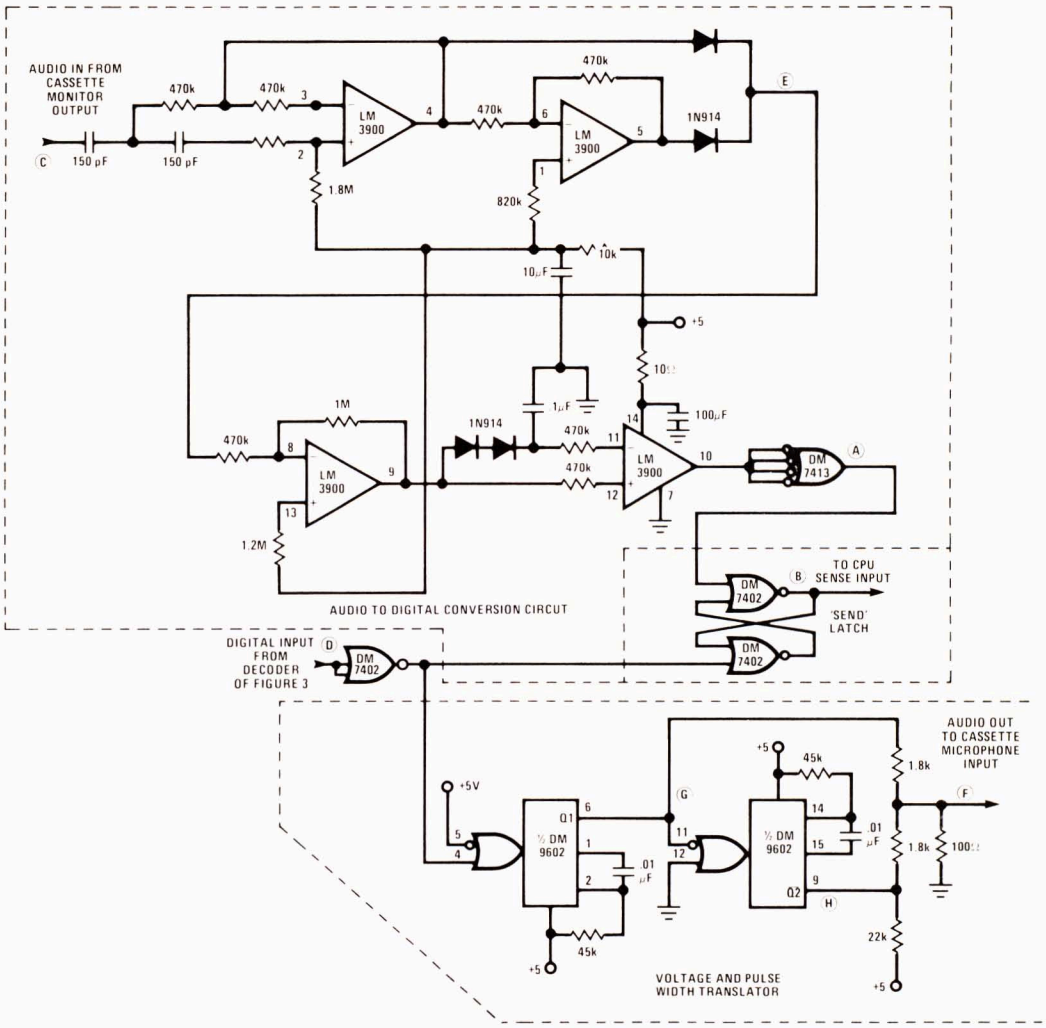


Figure 2. Interface Circuit

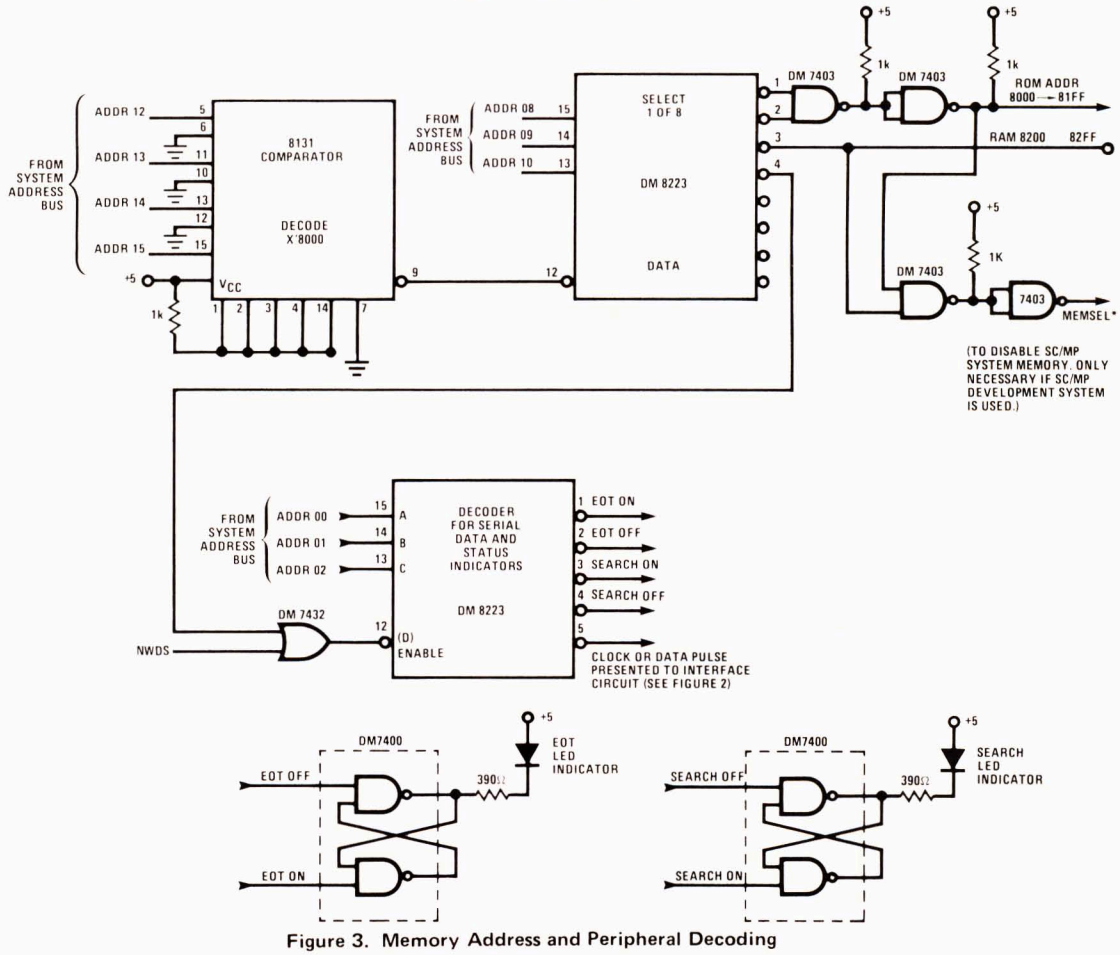


Figure 3. Memory Address and Peripheral Decoding

Table 1. Cassette Recorders Used

MAKE	MODEL	APPROX. COST
PANASONIC	RQ309AS RQ423S	$ 40 $ 70
SONY	TC-40 TC-55	$100 $155
To specify the ideal cassette for this application is difficult. The waveforms of figure 7C (output of cassette) may be used as a guide for determining reasonable characteristics.		

Table 2. Parts List

PART	QUANTITY
(Appl. BB. Card)	
4.375" X 4.862" P.C. Card	1
MM5204	1
MM2101	2
DM8131	1
DM8223	2
DM7403	1
DM7400	1
DM7432	1
DM7402	1
DM9602	1
LM3900	1
LM7413	1
Resistors (10%)	
470K	8
1 MEG	1
1.2 MEG	1
1.8 MEG	1
820K	1
45K	2
10K	1
22K	1
1.8K	2
100Ω	1
10Ω	1
1K	4
390Ω	2
Capacitors	
150 pF (MICA)	2
10 μF (Tantolum)	1
0.1 μF (ceramic)	11
0.01 μF (ceramic)	2
100 μF (tera T)	1
Diodes	
1N914	4

operation

A Tape I/O SC/MP Program residing in PROM (see appendix A for listing) provides the necessary timing and control for sending and receiving information between the SC/MP CPU and the cassette.

Typical Operation for Sending

1. Operator loads his program to be transmitted into RAM. This operation may be performed using a keyboard or a tape reader.
2. The recorder is turned on and the operator (using the keyboard entry) transfers control to the Data Write Routines, which output his program to the cassette through an interface circuit. Status indicators inform the operator when the transmission is complete.

Typical Operation for Receiving

1. Using the keyboard, the operator transfers control to the Receive Routine, which stores data transmitted from the cassette (through an interface circuit) into RAM.
2. The cassette is turned on and the operator observes status indicators to determine when the transmission is completed and whether or not all data have been received correctly.

functional description

Recording

Transmission from SC/MP to the cassette is accomplished using a scheme that is self-synchronizing on a bit-time basis.[1] The waveform is shown in figure 5. A 4-millisecond time frame is established by the 'send' routine. The time duration between clocks is data time. A logic 0 is represented by the absence of a pulse at the midpoint of the time frame; a logic '1', by the presence of a pulse.

The clocks and bit pulses are generated by the Address Decoder shown in figure 3. To generate either a clock or a bit pulse (for a logic '1'), a unique address is presented to the System Address Bus during the execution of a Store Instruction by SC/MP. The clock or pulse is then transmitted to the cassette via the Interface Circuit. A negative-going pulse is produced to begin the time frame. If the bit to be transmitted is a logic '1', the decoder is addressed at the midpoint of the time frame and a second negative-going pulse is produced again within one time frame at the decoder output. To transmit a logic '0', the decoder is not addressed at the midpoint of the time frame, and, thus, a logic '0' is indicated by the absense of the second negative-going pulse at the midpoint of the time frame.

To record, the Data Write Routines generate a long leader of zeros plus an identification word (see figure 6). The decoder presents these data to the Interface Circuit. The Voltage and Pulse-width Translator portion of the Interface Circuit changes the TTL signals from the Address Decoder into a form acceptable to the microphone input of the cassette (see figure 7A and 7B). The leader allows the tape-drive motor and AGC loop to stabilize and, also, is an interprogram gap that facilitates multiple-program recording on a single side of the cassette tape. User data are transmitted by SC/MP following the identification word. The data format is shown in figure 6.

[1] Scheme used was published in 'Computer Hobbyist' Volume 1, #5. **AUTHOR** – HAL CHAMBERLAIN

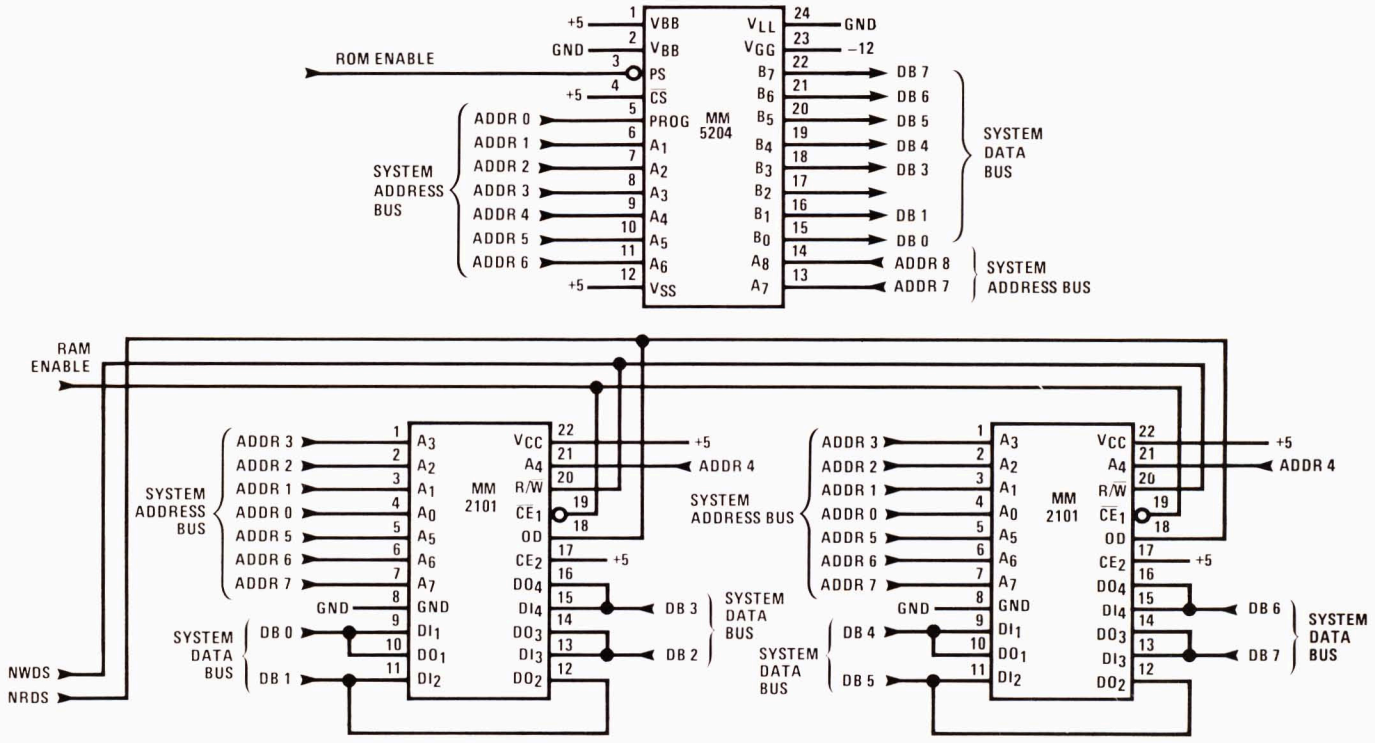


Figure 4. ROM/RAM Interconnection's Between SC/MP and Cassette Tape Recorder

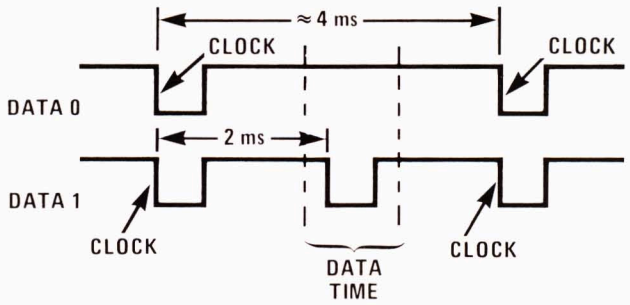


Figure 5. Data Output from Decoder

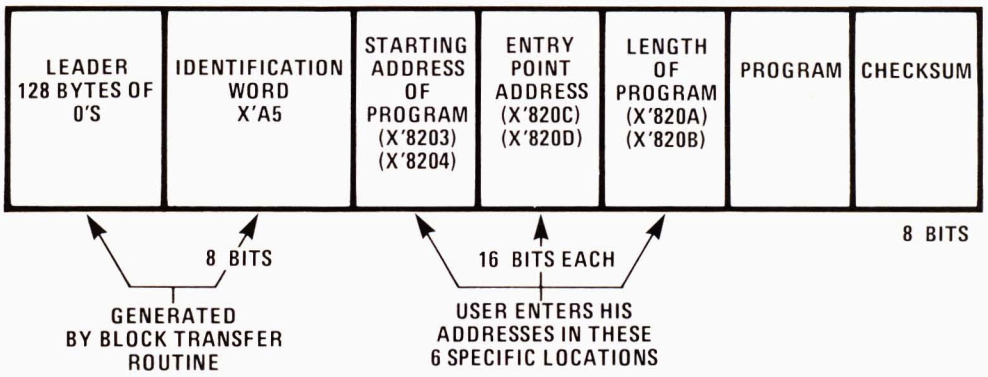


Figure 6. Message Format

Playback

When receiving data, the processor tests for a (logic '0') level at the sense B input. This level, from the 'send' latch in the Interface Circuit (see figure 2) indicates that the first pulse of the time frame has arrived. The latch then is reset by SC/MP after delaying one quarter of a time frame (see figure 7C, Bottom Trace). SC/MP delays an additional one half of a time frame and again tests for a logic '0' at the sense input; a second low level indicates that a 'logic 1' was received and the latch is then reset. Otherwise, absence of a second low level indicates that the data bit is a logic '0'. Using this method, a new time frame is generated by SC/MP upon receiving the first logic-'0' level of a time frame. Thus, no cumulative timing errors are built up. Refer to the figures 7C and 7D for waveforms that occur during the receive mode. These figures, which show transmissions of ones and zeros, are included as troubleshooting aids.

In the playback mode, the monitor output from the recorder is translated to a digital (TTL) signal by the Audio-to-Digital Conversion Circuit of figure 2; the TTL signal drives the sense B input of SC/MP. The Receive Routine searches for the identification word, and upon recognition stores the user data in RAM. Tape format is such that upon completion of loading a program from the cassette, the program may be executed or control may be transferred to another existing program (such as a debug program). This is accomplished by the user loading the Entry Point Address at record time with the starting location of any program desired. (This is discussed below under User Operation.) The Data Write and Receive Routines are stored in ROM along with a minimum control routine; these routines comprise the TAPE I/O SC/MP Routines of appendix A. The control routine effects communications between the operator and a "hex" keyboard and controls the LED indicators of figure 3.

user operation

Sending

NOTE

A hexadecimal number is identified by the prefix X'.

The operator may code a program into RAM using the keyboard. To input a program to the cassette, the operator loads location X'8203 (high-order byte) and X'8204 (low-order byte) with the starting 4-digit hexadecimal address of the program.

Next, location X'820C (high-order) and X'820D (low-order) are loaded with the hexadecimal address of the desired Entry Point. The entry point may be the starting address of any program to which the operator wants to transfer control upon completion of loading (playback mode). Finally, the length of the program is loaded into locations X'820A (high-order) and X'820B (low-order). To output the operator turns on the cassette and then transfers control to the address X'80C7, the beginning of the Data Write Routines. The Search Indicator is turned on after the leader of zeros is transmitted. When the transmission is completed, the Search Indicator is turned off, an End of Transmission Indicator is turned on, and the program halts at location X'8142.

Necessary communication between the operator and the processor must be provided so the operator may transfer control to a location in RAM and may modify RAM locations. This capability is available using the SC/MP development system or the SC/MP minimum DEBUG Kit plus Hex Keyboard, Interface and Hex Display.[2]

[2]Refer to SC/MP Technical Description, Appendix B.

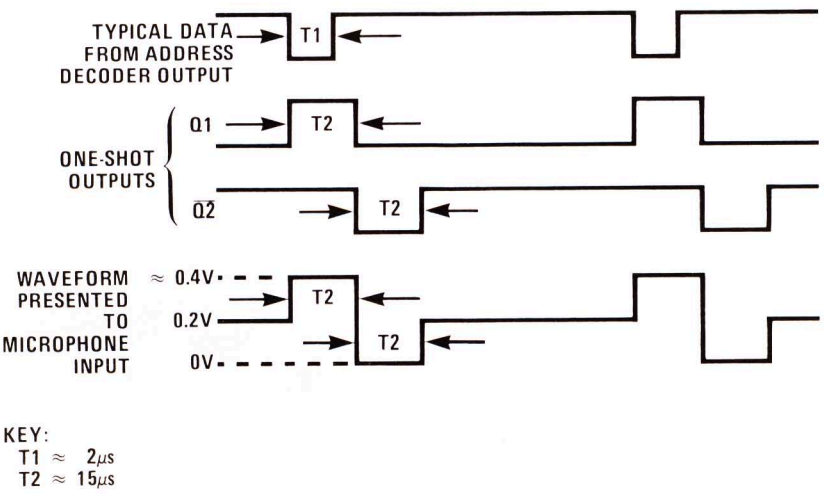


Figure 7A. Waveforms Developed by Voltage and Pulsewidth Circuit of Interface Circuit

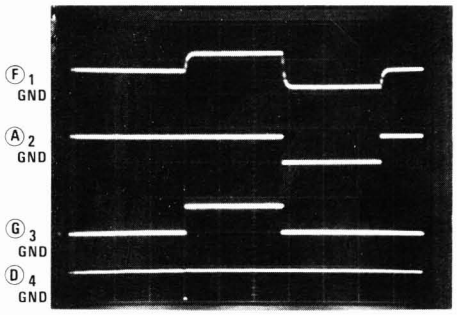


PHOTO NO. 1 – WRITE MODE

TRACE NO. 1 OUTPUT TO CASSETTE
VERTICAL = 0.5V/DIV
HORIZONTAL = 50 μs/DIV

TRACE NO. 2 ξ NO. 3 = ONE SHOT OUTPUTS
TRACE 2 = PIN 9 OF 9602 | SEE
TRACE 3 = PIN 6 OF 9602 | FIGURE ___
VERTICAL = 5V/DIV
HORIZONTAL = 50 μs/DIV

TRACE NO. 4 EFFECTIVE FLAG OUTPUT OF PERIPHERAL DECODER PIN 5
VERTICAL = 5V/DIV
HORIZONTAL = 10 μs/DIV

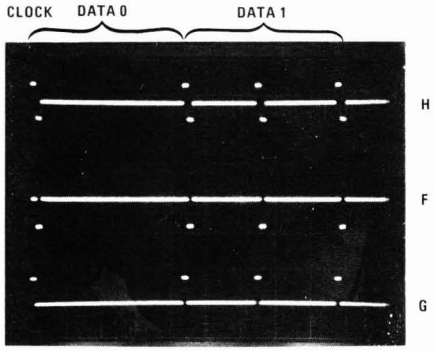


PHOTO NO. 2 – WRITE MODE

DATA OUT = ALTERNATE 0 ξ 1'S

TOP TRACE = OUTPUT TO CASSETTE
VERTICAL = 5V/DIV

MIDDLE TRACE = ONE SHOT PIN 9

BOTTOM TRACE = ONE SHOT PIN 6
VERTICAL = 5 V/DIV
HORIZONTAL = 1 μs/DIV
(ALL)

Note: Circled letters refer to points on Interface Circuit, Schematic Diagram.

Figure 7B. Write Mode Waveforms

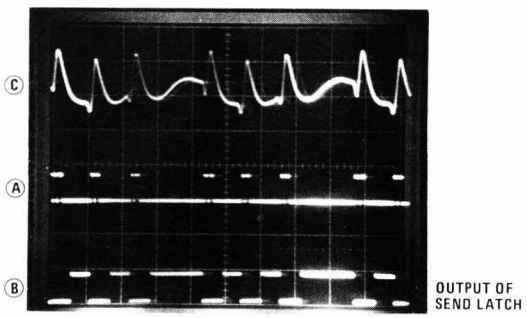


PHOTO NO. 3 – RECEIVE MODE

TOP TRACE = WAVEFORM FROM MONITOR OUTPUT OF RECORDER
MIDDLE TRACE = OUTPUT OF SCHMITT TRIGGER (7413 PIN 6)
BOTTOM TRACE = OUTPUT OF SEND F/F (TO SC/MP SENSE B INPUT)
VERTICAL = 5V/DIV
HORIZONTAL = 2 ms/DIV

Note: Circled letters refer to points on Interface Circuit, Schematic Diagram.

Figure 7C. Receive Mode Waveforms

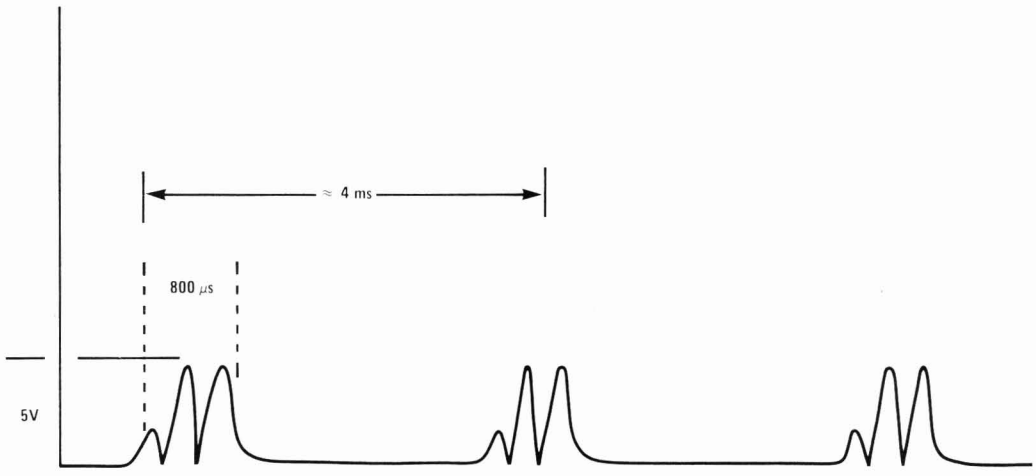


Figure 7D. Waveform at Point E of Interface Circuit

Receiving

To receive, using the keyboard, the operator transfers control to X'8000, the starting address of the Bootstrap Loader Routine, which initiates the required conditions in SC/MP and then addresses the Receive Routine. The operator turns on the cassette for 'PLAYBACK.' The EOT indicator is turned on at the end-of-transmission. The Search Indicator is turned on until the identification word is recognized and then is turned off. (This indicator will be on 3 to 5 seconds under normal operation.) If the checksum is good, the Search Indicator is turned on again when the transmission is completed.

When in the "playback" mode, the volume control of the cassette recorder should be adjusted to a point where the monitor output is just below "clipping" when measured with an oscilloscope. A trial and error method may be used using the Search Indicator on-off time of ~5 seconds as a limit switch while the volume control is varied.

Acknowledgements

The following people contributed to this application note:

Ed Burdick

Bill Grundman

Tom Harper

Tom Mills

References Publications

SC/MP Technical Description (Pub. No. 4200079)

SC/MP Users Manual (Pub. No. 4200105)

SC/MP Programming and Assembler Manual (Pub. No. 4200094)

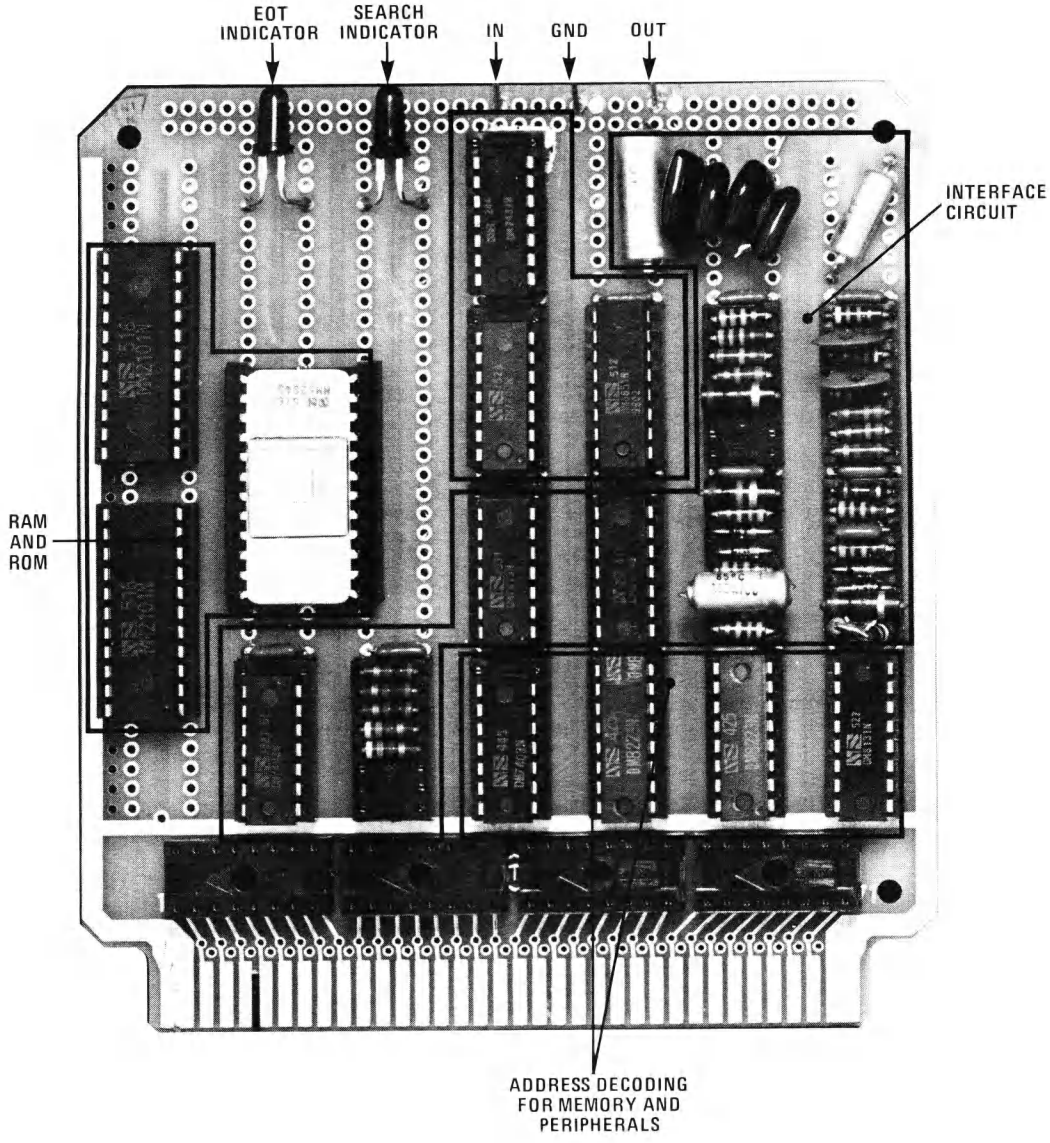


Figure 8. SC/MP-to-Cassette Breadboard

Appendix A
Tape IO SC/MP Routines

```
                        .TITLE  TAPEIO, ' SC/MP ROUTINES'

        8000            .=X'8000

        8200  RAM       =       X'8200          ; RAM ADDRESS FOR POINTER
        8300  PERIPH    =       X'8300          ; PERIPHERAL ADDRESS FOR POINT

        0003  P3        =       3               ; POINTER #3
        0002  P2        =       2               ; POINTER #2
        0001  P1        =       1               ; POINTER #1

              ; TEMPORARY DATA IN RAM

        0000  CNTU      =       0               ; INSIDE COUNTER FOR LEADER
        0001  CNTL      =       1               ; OUTSIDE COUNTER FOR LEADER
        0002  CKSUM     =       2               ; CHECK SUM COUNTER
        0003  STARTU    =       3               ; STARTING ADDRESS (UPPER)
        0004  STARTL    =       4               ; STARTING ADDRESS (LOWER)
        0005  BITCNT    =       5               ; BIT COUNTER
        0006  TEMP1     =       6               ; TEMPORARY STORAGE LOCATIONS
        0007  TEMP2     =       7               ;      "        "        "
        0008  TEMP3     =       8               ;      "        "        "
        0009  TEMP4     =       9               ;      "        "        "
        000A  WDCNTU    =       10              ; WORD COUNT (UPPER)
        000B  WDCNTL    =       11              ; WORD COUNT (LOWER)
        000C  JUMPU     =       12              ; TRANSFER ADDRESS (UPPER)
        000D  JUMPL     =       13              ; TRANSFER ADDRESS (LOWER)

              ; PERIPHERAL ORDER CODES

        0000  EOTON     =       0               ; END OF TAPE LED ON POINTER
        0001  EOTOFF    =       1               ; END OF TAPE LED OFF POINTER
        0002  SRCHON    =       2               ; SEARCH LED ON POINTER
        0003  SRCHOF    =       3               ; SEARCH LED OFF POINTER
        0004  FLAG      =       4               ; READ/WRITE FLAG


                        .PAGE 'BOOTSTRAP LOADER'


              ; BOOTSTRAP LOADER ROUTINE.  RECEIVES PROGRAM FROM TAPE.
              ; ALL NECCESSARY INFORMATION FOR LOADING IS ON TAPE.

              ; THIS PROGRAM MAY BE REASSEMBLED TO ADDRESS X'0000 TO
              ; FUNCTION AS A POWER-ON LOADER.

              ; INPUT OPERATION OF LED INDICATORS:

              ;       SEARCH LED ON WHEN PROGRAM STARTS
              ;       SEARCH LED OFF WHEN IDENTIFIER CHARACTER RECEIVED
              ;       END OF TAPE (EOT) LED ON WHEN RECEPTION COMPLETE
              ;       SEARCH LED ON IF CHECKSUMS COMPARE

              ;       CONTROL IS THEN TRANSFERED TO USER PROGRAM


8000 08     BOOT:   NOP                     ; FOR RELOCATION TO X'0000
8001 C400           LDI     L(RAM)          ; INITIALIZE RAM POINTER
8003 32             XPAL    P2              ; IN P2
8004 C482           LDI     H(RAM)
8006 36             XPAH    P2
8007 C400           LDI     0               ; CLEAR ACCUMULATOR
8009 CA02           ST      CKSUM(P2)       ; INITIALIZE CHECKSUM COUNTER
800B C400           LDI     L(PERIPH)       ; PUT PERIPHERAL POINTER
800D 33             XPAL    P3              ; IN P3
800E C483           LDI     H(PERIPH)
8010 37             XPAH    P3
8011 CB02           ST      SRCHON(P3)      ; TURN ON SEARCH LED
8013 CB01           ST      EOTOFF(P3)      ; TURN OFF END OF TAPE LED
8015 C400           LDI     0               ; CLEAR ACCUMULATOR
8017 01             XAE                     ; CLEAR E REGISTER
8018 C48F           LDI     L(GETBIT)-1     ; PLACE ADDRESS OF GET BIT
801A 31             XPAL    P1              ; IN P1
801B C480           LDI     H(GETBIT)
801D 35             XPAH    P1
801E 3D     LOCID:  XPPC    P1              ; GO TO GETBIT FOR INPUT
801F 40             LDE
8020 E4A5           XRI     X'A5            ; CHECK FOR PROPER ID CHARACTE
8022 9802           JZ      SETPNT          ; IF ID RECEIVED, TAKE REST OF
8024 90F8           JMP     LOCID           ; PROGRAM, ELSE GET NEXT BIT
```

```
 80 8026 CB03  SETPNT: ST      SRCHOF(P3)      ; TURN OFF SEARCH LED
 81 8028 C46D          LDI     L(RECV)-1       ; PLACE ADDRESS OF BYTE RECEIV
 82 802A 31            XPAL    P1              ; IN P1
 83 802B C480          LDI     H(RECV)
 84 802D 35            XPAH    P1
 85 802E 3D            XPPC    P1              ; GET STARTING ADDRESS (LOWER)
 86 802F 33            XPAL    P3              ; AND PLACE IN P3
 87 8030 3D            XPPC    P1              ; GET STARTING ADDRESS (UPPER)
 88 8031 37            XPAH    P3
 89 8032 3D            XPPC    P1              ; GET TRANSFER ADDRESS AND
 90 8033 CA0D          ST      JUMPL(P2)       ; SAVE IN RAM
 91 8035 3D            XPPC    P1
 92 8036 CA0C          ST      JUMPU(P2)
 93 8038 3D            XPPC    P1              ; GET WORD COUNT (LOWER)
 94 8039 CA0B          ST      WDCNTL(P2)
 95 803B 3D            XPPC    P1              ; GET WORD COUNT (UPPER)
 96 803C CA0A          ST      WDCNTU(P2)
 97
 98 803E 3D    BOOTIN: XPPC    P1              ; GO TO RECEIVE
 99 803F CF01          ST      @1(P3)          ; STORE AND INCREMENT POINTER
100 8041 F202          ADD     CKSUM(P2)       ; ADD CHARACTER TO CHECKSUM
101 8043 CA02          ST      CKSUM(P2)
102 8045 AA0B          ILD     WDCNTL(P2)      ; INCREMENT LOWER WORD COUNTER
103 8047 9CF5          JNZ     BOOTIN          ; CHECK FOR ZERO
104 8049 AA0A          ILD     WDCNTU(P2)      ; INCREMENT UPPER WORD COUNTER
105 804B 9CF1          JNZ     BOOTIN          ; CHECK FOR END OF TRANSMISSIO
106 804D 3D            XPPC    P1              ; GET CHECKSUM FROM TAPE
107 804E E202          XOR     CKSUM(P2)       ; COMPARE TO CALCULATED VALUE
108 8050 9809          JZ      EXECPR          ; EXECUTE LOADED PROGRAM
109 8052 C400          LDI     L(PERIPH)
110 8054 33            XPAL    P3
111 8055 C483          LDI     H(PERIPH)
112 8057 37            XPAH    P3
113 8058 CB00          ST      EOTON(P3)       ; TURN ON EOT LED TO INDICATE
114 805A 00            HALT                    ; CHECKSUM ERROR AND HALT
115
116 805B C400  EXECPR: LDI     L(PERIPH)
117 805D 33            XPAL    P3
118 805E C483          LDI     H(PERIPH)
119 8060 37            XPAH    P3
120 8061 CB00          ST      EOTON(P3)       ; TURN ON END OF TAPE LED
121 8063 CB02          ST      SRCHON(P3)      ; TURN ON SEARCH LED
122 8065 C20D          LD      JUMPL(P2)       ; LOAD TRANSFER ADDRESS
123 8067 33            XPAL    P3
124 8068 C20C          LD      JUMPU(P2)
125 806A 37            XPAH    P3
126 806B C7FF          LD      @-1(P3)         ; DECREMENT POINTER FOR FETCH
127 806D 3F            XPPC    P3              ; EXECUTE
128
129
130             ; RECEIVE ROUTINE.  RECEIVES ONE 8-BIT CHARACTER INTO
131             ; ACCUMULATOR.
132
133
134 806E C48F  RECV:   LDI     L(GETBIT)-1     ; PLACE ADDRESS OF GETBIT
135 8070 31            XPAL    P1              ; IN P1
136 8071 CA07          ST      TEMP2(P2)       ; SAVE CURRENT CONTENTS OF P1
137 8073 C480          LDI     H(GETBIT)
138 8075 35            XPAH    P1
139 8076 CA06          ST      TEMP1(P2)
140 8078 C408          LDI     8               ; SET BIT COUNT
141 807A CA05          ST      BITCNT(P2)
142 807C C400          LDI     0               ; CLEAR ACCUMULATOR
143 807E 01            XAE                     ; CLEAR E REGISTER
144 807F 3D    LOOP:   XPPC    P1              ; GO TO GETBIT
145 8080 BA05          DLD     BITCNT(P2)      ; DECREMENT BIT COUNT
146 8082 9802          JZ      RETRN2          ; CHECK FOR ZERO
147 8084 90F9          JMP     LOOP
148 8086 C207  RETRN2: LD      TEMP2(P2)       ; RESTORE P1 TO ORIGINAL
149 8088 31            XPAL    P1              ; CONTENTS
150 8089 C206          LD      TEMP1(P2)
151 808B 35            XPAH    P1
152 808C 40            LDE                     ; PLACE CHARACTER IN ACC.
153 808D 3D            XPPC    P1              ; RETURN
154 808E 90DE          JMP     RECV
155
156
157             ; GET BIT ROUTINE.  RECEIVES 1 BIT INTO E REGISTER
158
```

```
159
160 8090 C400  GETBIT: LDI     L(PERIPH)       ; PLACE PERIPHERAL ADDR.  IN P3
161 8092 33           XPAL    P3
162 8093 CA09         ST      TEMP4(P2)       ; SAVE ORIGINAL CONTENTS OF P3
163 8095 C483         LDI     H(PERIPH)
164 8097 37           XPAH    P3
165 8098 CA08         ST      TEMP3(P2)
166 809A 19           SIO                     ; SHIFT E REGISTER
167 809B 06    CKSA:  CSA                     ; COPY STATUS TO ACCUMULATOR
168 809C D420         ANI     X'20            ; MASK
169 809E 9802         JZ      CLOCK           ; IF ZERO, BIT RECEIVED
170 80A0 90F9         JMP     CKSA            ; CHECK AGAIN
171 80A2 C400  CLOCK: LDI     0               ; CLEAR ACCUMULATOR FOR DELAY
172 80A4 8F01         DLY     1               ; DELAY 1 MS (1/4 BIT TIME)
173 80A6 CB04         ST      FLAG(P3)        ; RESET LATCH
174 80A8 C400         LDI     0               ; INIT ACCUMULATOR FOR DELAY
175 80AA 8F02         DLY     2               ; DELAY PAST MIDDLE OF WINDOW
176 80AC 06           CSA                     ; COPY STATUS TO ACCUMULATOR
177 80AD D420         ANI     X'20            ; MASK
178 80AF 9802         JZ      ONE             ; IF ZERO, THEN BIT IS A "1"
179 80B1 9004         JMP     RESET
180 80B3 40    ONE:   LDE
181 80B4 DC80         ORI     X'80            ; ADD "1" BIT TO CHARACTER
182 80B6 01           XAE                     ; SAVE IN E REGISTER
183 80B7 CB04  RESET: ST      FLAG(P3)        ; RESET LATCH
184 80B9 06           CSA                     ; COPY STATUS TO ACCUMULATOR
185 80BA D420         ANI     X'20            ; MASK
186 80BC 98F9         JZ      RESET           ; CHECK IF LATCH IS RESET
187 80BE C209  RETRN3: LD      TEMP4(P2)       ; RESTORE P2
188 80C0 33           XPAL    P3
189 80C1 C208         LD      TEMP3(P2)
190 80C3 37           XPAH    P3
191 80C4 3D           XPPC    P1              ; RETURN
192 80C5 90C9         JMP     GETBIT
193
194
                      .PAGE 'DATA WRITE ROUTINES'
195
196
197            ; SEND 4 SECONDS OF "0" (ABOUT 1000) TO ALLOW FOR
198            ; TAPE TO SETTLE ON PLAY BACK AND ACT AS LEADER
199
200            ; OUTPUT OPERATION OF LED INDICATORS:
201
202            ;       SEARCH LED ON WHEN LEADER COMPLETE
203            ;       SEARCH LED OFF WHEN TRANSMISSION COMPLETE
204            ;       END OF TAPE LED ON WHEN TRANSMISSION COMPLETE
205
206
207 80C7 C400  INIT:   LDI     L(RAM)          ; PLACE RAM POINTER IN P2
208 80C9 32           XPAL    P2              ;       (LOWER)
209 80CA C482         LDI     H(RAM)
210 80CC 36           XPAH    P2              ;       (UPPER)
211 80CD C400         LDI     L(PERIPH)       ; PLACE PERIPHERAL ADDRESS
212 80CF 33           XPAL    P3              ; IN P3
213 80D0 C483         LDI     H(PERIPH)
214 80D2 37           XPAH    P3
215 80D3 C400  COMP:  LDI     0               ; CLEAR ACCUMULATOR
216 80D5 02           CCL                     ; CLEAR CARRY/LINK FLAG
217 80D6 FA0B         CAD     WDCNTL(P2)      ; FORM 1'S COMP OF LOWER COUNT
218 80D8 CA0B         ST      WDCNTL(P2)
219 80DA C400         LDI     0               ; CLEAR ACCUMULATOR
220 80DC FA0A         CAD     WDCNTU(P2)      ; FORM 1'S COMP OF UPPER COUNT
221 80DE CA0A         ST      WDCNTU(P2)
222 80E0 CB03         ST      SRCHOF(P3)      ; TURN OFF SEARCH LED
223 80E2 CB01         ST      EOTOFF(P3)      ; TURN OFF END OF TAPE LED
224 80E4 C408  SNDLDR: LDI     8               ; SET OUTER COUNTER
225 80E6 CA01         ST      CNTL(P2)
226 80E8 C480  CNT1:  LDI     X'80            ; SET INNER COUNTER
227 80EA CA00         ST      CNTU(P2)
228 80EC CB04  CNT2:  ST      FLAG(P3)        ; PULSE WRITE FLAG
229 80EE C400         LDI     0               ; CLEAR ACCUMULATOR
230 80F0 8F04         DLY     4               ; DELAY 1 BIT TIME
231 80F2 BA00         DLD     CNTU(P2)        ; DECREMENT INNER COUNTER
232 80F4 9CF6         JNZ     CNT2            ; CHECK FOR ZERO
233 80F6 BA01         DLD     CNTL(P2)        ; DECREMENT OUTER COUNTER
234 80F8 94EE         JP      CNT1            ; CHECK FOR LESS THEN ZERO
235 80FA CB02         ST      SRCHON(P3)      ; TURN ON SEARCH LED
236
```

```
237
238                ; BLOCK TRANSFER ROUTINE.  SENDS BLOCK OF DATA TO CASETTE
239
240                ; THE FOLLOWING ADDRESSES MUST BE LOADED BY USER BEFORE
241                ; EXECUTING THE WRITE PROGRAM:
242
243                ; X'8203 -- UPPER 8 BITS OF PROGRAM ADDRESS
244                ; X'8204 -- LOWER 8 BITS OF PROGRAM ADDRESS
245                ; X'820A -- UPPER 8 BITS OF PROGRAM LENGTH
246                ; X'820B -- LOWER 8 BITS OF PROGRAM LENGTH
247                ; X'820C -- UPPER 8 BITS OF TRANSFER ADDRESS (ENTRY POINT)
248                ; X'820D -- LOWER 8 BITS OF TRANSFER ADDRESS
249
250
251 80FC C400  BLOCK:  LDI     0                ; CLEAR ACCUMULATOR
252 80FE CA02          ST      CKSUM(P2)        ; INITIALIZE CHECKSUM COUNTER
253 8100 C481          LDI     H(WRITE)         ; PLACE ADDRESS OF WRITE IN P1
254 8102 35            XPAH    P1
255 8103 C444          LDI     L(WRITE)-1
256 8105 31            XPAL    P1
257 8106 C4A5          LDI     X'A5             ; LOAD ACCUMULATOR WITH ID
258 8108 3D            XPPC    P1               ; WRITE ID ON TAPE
259 8109 C204          LD      STARTL(P2)       ; GET STARTING ADDRESS
260 810B 3D            XPPC    P1               ; WRITE ONTO TAPE
261 810C C203          LD      STARTU(P2)
262 810E 3D            XPPC    P1
263 810F C20D          LD      JUMPL(P2)        ; GET TRANSFER ADDRESS
264 8111 3D            XPPC    P1
265 8112 C20C          LD      JUMPU(P2)
266 8114 3D            XPPC    P1
267 8115 C20B          LD      WDCNTL(P2)       ; GET LENGTH
268 8117 3D            XPPC    P1
269 8118 C20A          LD      WDCNTU(P2)
270 811A 3D            XPPC    P1
271 811B C204  GETBYT: LD      STARTL(P2)       ; PLACE CURRENT ADDRESS IN P1
272 811D 31            XPAL    P1
273 811E C203          LD      STARTU(P2)
274 8120 35            XPAH    P1
275 8121 C501          LD      @1(P1)           ; GET CHARACTER THROUGH
276 8123 01            XAE                      ; POINTER AND SAVE IN E REG.
277 8124 C444          LDI     L(WRITE)-1       ; GET ADDRESS OF WRITE AND
278 8126 31            XPAL    P1               ; SAVE CURRENT CONTENTS OF P1
279 8127 CA04          ST      STARTL(P2)
280 8129 C481          LDI     H(WRITE)
281 812B 35            XPAH    P1
282 812C CA03          ST      STARTU(P2)
283 812E 40            LDE
284 812F F202          ADD     CKSUM(P2)        ; UPDATE CHECKSUM
285 8131 CA02          ST      CKSUM(P2)
286 8133 40            LDE                      ; PLACE CHARACTER IN ACC.
287 8134 3D            XPPC    P1               ; SEND CHARACTER
288 8135 AA0B          ILD     WDCNTL(P2)       ; INCREMENT WORD COUNTER
289 8137 9CE2          JNZ     GETBYT           ; CHECK FOR ZERO
290 8139 AA0A          ILD     WDCNTU(P2)
291 813B 9CDE          JNZ     GETBYT
292 813D C202          LD      CKSUM(P2)        ; SEND CHECKSUM TO TAPE
293 813F 3D            XPPC    P1
294 8140 CB03          ST      SRCHOF(P3)       ; TURN OFF SEARCH LED
295 8142 CB00          ST      EOTON(P3)        ; TURN ON END OF TAPE LED
296 8144 00            HALT                     ; HALT WHEN FINISHED
297
298
299                ; DATA WRITE ROUTINE.  WRITES 1 8-BIT CHARACTER ON TAPE
300
301
302 8145 01    WRITE:  XAE                      ; SAVE CHARACTER IN E REG.
303 8146 C408          LDI     8                ; SET BIT COUNT
304 8148 CA05          ST      BITCNT(P2)
305 814A 40    MASK:   LDE
306 814B D401          ANI     1                ; MASK
307 814D 9C08          JNZ     SEND1            ; CHECK IF BIT "0" OR "1"
308 814F C400          LDI     0                ; CLEAR ACCUMULATOR FOR DELAY
309 8151 CB04  SEND0:  ST      FLAG(P3)         ; PULSE WRITE FLAG
310 8153 8F04          DLY     4                ; DELAY 1 BIT TIME ( 4 MS )
311 8155 900C          JMP     SHIFT
312 8157 C400  SEND1:  LDI     0
313 8159 CB04          ST      FLAG(P3)         ; PULSE WRITE FLAG
314 815B 8F02          DLY     2                ; DELAY TO MIDDLE OF WINDOW
```

Tape IO SC/MP Routines (cont'd.)

```
315 815D CB04           ST      FLAG(P3)          ; PULSE WRITE FLAG
316 815F C400           LDI     0                 ; CLEAR ACC.  FOR DELAY
317 8161 8F02           DLY     2                 ; DELAY TO END OF WINDOW
318
319 8163 19     SHIFT:  SIO                       ; SHIFT E REGISTER
320 8164 BA05           DLD     BITCNT(P2)        ; DECREMENT BIT COUNTER
321 8166 9802           JZ      RETRN1            ; CHECK FOR ZERO
322 8168 90E0           JMP     MASK              ; SEND NEXT BIT
323 816A 3D     RETRN1: XPPC    P1                ; RETURN
324 816B 90D8           JMP     WRITE
325
326       8000            END     BOOT
```

```
BITCNT  0005          BLOCK   80FC *        BOOT    8000
BOOTIN  803E          CKSA    809B          CKSUM   0002
CLOCK   80A2          CNT1    80E8          CNT2    80EC
CNTL    0001          CNTU    0000          COMP    80D3 *
EOTOFF  0001          EOTON   0000          EXECPR  805B
FLAG    0004          GETBIT  8090          GETBYT  811B
INIT    80C7 *        JUMPL   000D          JUMPU   000C
LOCID   801E          LOOP    807F          MASK    814A
ONE     80B3          P1      0001          P2      0002
P3      0003          PERIPH  8300          RAM     8200
RECV    806E          RESET   80B7          RETRN1  816A
RETRN2  8086          RETRN3  80BE *        SEND0   8151 *
SEND1   8157          SETPNT  8026          SHIFT   8163
SNDLDR  80E4 *        SRCHOF  0003          SRCHON  0002
STARTL  0004          STARTU  0003          TEMP1   0006
TEMP2   0007          TEMP3   0008          TEMP4   0009
WDCNTL  000B          WDCNTU  000A          WRITE   8145
```

```
NO ERROR LINES
SOURCE CHECKSUM=C694
FIRST INPUT SECTOR HEX -  03DD
FINAL INPUT SECTOR HEX -  03EE
```

Manufactured under one or more of the following U.S. patents: 3083262, 3189758, 3231797, 3303356, 3317671, 3323071, 3381071, 3408542, 3421025, 3426423, 3440498, 3518750, 3519897, 3557431, 3560765, 3566218, 3571630, 3575609, 3579059, 3593069, 3597640, 3607469, 3617859, 3631312, 3633052, 3638131, 3648071, 3651565, 3693248.

National Semiconductor Corporation
2900 Semiconductor Drive, Santa Clara, California 95051, (408) 737-5000/TWX (910) 339-9240
National Semiconductor GmbH
808 Fuerstenfeldbruck, Industriestrasse 10, West Germany, Tele. (08141) 1371/Telex 05-27649
National Semiconductor (UK) Ltd.
Larkfield Industrial Estate, Greenock, Scotland, Tele. (0475) 33251/Telex 778-632

National does not assume any responsibility for use of any circuitry described, no circuit patent licenses are implied, and National reserves the right, at any time without notice, to change said circuitry.

COMPUTE LIBRARY SUBMITTAL FORM

☐ SC/MP ☐ PACE ☐ IMP-16 ☐ OTHER _____

☐ PACER ☐ SCAMPER

Program Title

Function

Required Hardware

Required Software

Input Parameters

Output Results

(use additional sheets if necessary)

Registers Modified: _____

Maximum Subroutine Nesting Level: _____

RAM Required: _____

Assembler/Compiler Used: _____

ROM Required: _____

Programmer: _____

Date Submitted: _____

Company: _____

1. Complete Submittal Form as follows:

   a. Processor (check appropriate box)
   b. Program title: Name or brief description of program function
   c. Function: description of operations performed by the program
   d. Required hardware/firmware/software
      For example: TTY
      High Speed Printer
      Arithmetic CROM
      POWR I/O CROM
      EXTENDED CROM
      TTY routines
      Floating point package
      Support software required for cross products
   e. Input parameters: Description of register values, memory areas or values accepted from input ports
   f. Output results: Values to be expected in registers, memory areas or on output ports
   g. Program details
      1. Registers modified
      2. RAM required (bytes)
      3. ROM required (bytes)
      4. Maximum subroutine nexting level
   h. Assembler/Compiler used
      For example: SM/PL
      IMPASM, PASM
      PACE CROSS ASSEMBLER
      SC/MP CROSS ASSEMBLER
   i. Programmer and company

2. A source listing of the program and paper tape should be included

3. A test program which assures the validity of the contributed program is useful to include for user. This is for the user's verification:

   Each library program submitted entitles you or one of your colleagues to a free membership.

   Name (add Address for free COMPUTE membership)

Name_____ _____

Company_____ _____

SEND COMPLETED FORMS WITH PROGRAM LISTINGS AND SOURCE TAPES TO:

COMPUTE/115
National Semiconductor
2900 Semiconductor Dr.
Santa Clara, CA 95051
ATTN: Georgia Marszalek

OR

National Semiconductor Corp. Bmbh
808 Fuerstenfeldbruck
Industriestrasse 10
Germany
ATTN: Phil Hughes

Support Hardware for SC/MP and PACE Application Cards!

Bob Pecotich, National Semiconductor

SC/MP and PACE share a common 4.375" X 4.862" card size, with both families of application cards using the standard 72 pin, 0.1 with center edge connector. The card size is one widely supported by *AUGAT, INC.* (with its M Series line) and a number of second sources. The following table lists support hardware commonly needed by your customers . . .

Manufacturer	Part No.	Price
72 CONTACT EDGE CONNECTOR		
Augat	14005-17P3	$4.81/1, $4.22/10-24
obinson/Nugent	EC721	$7.55/1, $5.20/100
Elco	00-6307-072-309-001	
Cerich	50-72C-30	
National Connector	900100-36	
Stanford Applied Engineering	CDP7000-72	
Winchester	HW36C0111	$4.79/1, $3.51/100
CONNECTOR CARD CAGE WITH BACKPLANE:		
Augat	8170-MG1	$177.50/1, $147.75/10-24
Robinson/Nugent	MECA-1	$202/1, $150/10-24
EXTENDER CARD:		
Augat	8136-MG13	$29.50/1, $24.50/10-24
Robinson/Nugent	EB-72	$31.60/1
NIVERSAL W/W CARD ITH TERMINALS:		
Augat	8136-UMG1	$51.75/1
Robinson/Nugent	MAL-UNI-24	$48.60/1
HIGH DENSITY W/W CARD WITH TERMINALS:		
Augat	8136-MG-15	$85.25/1, $71.50/10

NEW ADDRESS FOR MICROPROCESSOR SERVICE CENTER

Microprocessor Service Center
National Semiconductor
2921 Copper Rd.
Santa Clara, CA 95051

For information regarding the status of equipment or service policies call (408) 737-6270. Ask for Jim Snyder or Don Cooper.

CONFERENCE ANNOUNCEMENTS

MINI/MICROCOMPUTER CONFERENCE AND EXPOSITION
October 19-20-21, 1976
Brooks Hall/Civic Auditorium, San Francisco

A Major Computer Conference in a Major Computer Market

Approximately twenty sessions consisting of eighty pages covering both application and design topics are planned.

PLUS . . . tutorial sessions on minis and micros and a special session for computer hobbyists!

For information, write:

MINI/MICRO COMPUTER CONFERENCE AND EXPOSITION
5544 E. LaPalma Avenue, Anaheim, CA 92807
Phone: (714) 528-2400

MICRO-9

Ninth Annual Workshop on Microprogramming to be held in New Orleans, *September 27-29*. For information contact Prof. Peter Kornerup, MICRO-9 Program Chairman, Computer Science Dept., University of Southwestern Louisiana, Box 4-4330, Lafayette, LA 70504 (318) 233-3850, Ext. 538.

Not Free, But Affordable!

National Semi Handbooks. The eight-bit *SC/MP Technical Description* starts with a general introduction for nontechnical users and follows up with complete details of the design of SC/MP-based applications. Price for the 65-page handbook is $3 . . . The *Memory Data Book* covers most of National's memory and memory-related products including bipolar, MOS, CMOS RAMs and PROMs. Price for the 544-page book is $3 . . . The *TTL Data Handbook* describes National's complete line of bipolar logic devices. A tri-state selection guide, industry cross-reference guide and functional index are also included. Price is $4 . . . The 16-bit *Pace Technical Description* describes both the full-feature CPU and the entire complement of hardware and software items. Price for the 96-page handbook is $3.

To obtain these handbooks, send check for amount (California residents add 6 percent sales tax) to *Marketing Services Dept., National Semiconductor Corp., 2900 Semiconductor Drive, Santa Clara, CA 95051.*

ERRATA

The September 1975 issue of the PACE Data Book contains the following corrections:

Page 23	The PACE ILE/16 (IPC-168/513J) has been replaced by the PACE ILE/8 (IPC-16A/503J), and this latter part should be used in new designs.
Page 40	The ALE/8 (IPC-16A/508J) has been replaced by the ALE/16 (IPC-16A/518J), and this latter part should be used in new designs.
Pages 45, 46	The "J" package in which the Blue/Green Chips are currently supplied is not a hermetic cavity DIP as described. Instead, the package consists of a ceramic substrate to which the chips are fixed. The attached chips are protected by a conformal epoxy coating that provides hermeticity comparable to an Epoxy B package.

SC/MP, PACE TRAINING AT EUROPEAN WORKSHOPS & SEMINARS!

SC/MP WORKSHOPS

Belgium
Date: September 13-15 inclusive
Location: Brussels
Sponsor: J.P. Lemaire
Rame Galoise 1 A
1020 Brussels
Telephone: (02)-478.48.47
Telex: 24.612

Norway
Date: October 11-13
Location: Oslo
Further information from:
NS Sweden 0046-8-970835

Finnland
Date: September 28-30
Location: Helsinki
Further information from:
NS Sweden
Multikomponent
Copenhagen

Denmark
Date: September 1-3 inclusive
Location: Copenhagen
Sponsor: Multikomponent
Herstedvang 7 C
2020 Albertslund
Denmark
Telephone: (02)-644477

France
Dates: September 20-24, November 2-26
Location: Paris
Sponsor: Fime
3 Rue de Chevilly
94262 Fresnes, France
Telephone: 666 95 01
Telex: 204802

Switzerland
Date: September 27-29
Location: Geneva
Sponsor: Fenner
Rheinfelderstr 16-18
4450 Sissach
Switzerland
Telephone: (061)-982202
Telex: 63235

U.K.
1. Date: September 6
Location: Birmingham
2. Date: October 4
Location: Manchester
3. Date: November 1
Location: London
4. Date: November 29
Location: Leeds

(Continued on page 12)

the Bit·Bucket

Dear Sirs:

Enclosed you will find my application for membership in COMPUTE. I am interested in obtaining a list of the routines currently in the User's Group Library. Specifically, I am interested in obtaining a routine to calculate the square root of a positive integer number.

Sincerely,
David L. Wilson
Air Monitoring Inc.
2015 Bellaire Avenue
Royal Oak, Michigan 48067

Mr. Wilson has a PACE kit and a PACER system. We have a SQRT program for IMP-16 (SL0028A). Does anyone have one for PACE? Ed.

16 BIT COMPUTER KIT (Cont'd)

The keyboard will also allow you to sequentially increment or decrement through memory or internal accumulators and registers for examination or modification of contents. Other front panel buttons include run, initialize (reset), restart (halt CPU but do not reset), and cancel last command.

The kit includes all parts one needs to have a working desk top microcomputer development system. The CPU board includes the PACE 16 bit MPU with necessary input and output buffers. On the control and I/O boards are two DM8531 (2038 X 8 each) ROMs for the system monitor. Also included on these boards are four MM2112 (256 X 4) static RAMs, one MM5740 keyboard encoder, two hex latches and LED driver circuits as well as all required support components to interface with the two 4 digit displays and 32 keypad. The control board has space for four more MM2112 RAMs. The memory board comes with four MM2112 RAMs. Space is provided for 12 more MM2112 RAMs and four MM5204 (512 X 8) PROMs for future memory expansion. The PAC II card has 2k X 16 of MOS RAM. The PAC I, PAC II, and PAC III cards are optional cards. PAC III is a prototyping card with voltage regulators. All other boards also have their own on board voltage regulators. PAC I is a TTY (or RS232C) interface and resident assembler card. With PAC I the user may perform all the front panel functions from a teletype (or similar device using current loop or RS232C) as well as the following useful functions:

- Load or punch a paper tape — no bootstrap need be loaded.
- Display a block of memory in one of several formats including assembly language (yes a dis-assembler! — very useful), ASCII, hexadecimal, unsigned decimal, or signed decimal.
- Set, list, or reset break or snap points (Break points are placed at strategic locations in a program. They halt execution and display the contents of specified registers and memory locations. Snap points do the same except program execution is not halted.)
- Enter programs in assembly language format (the assembler converts your programs line by line as you type them, to hexadecimal. No paper tape or cassette need be used for this. The assembler and other features listed here reside in two EA4900 type ROMs which hold 16k bits each.)

- Use symbols; the assembler does all address assignment and referencing. (One may also list the symbol table, delete a symbol or clear the table.)

The Pacer worked perfectly the first time I turned on the power. As I played with it, I began to appreciate the beauty of its high level front panel operational and debug capabilities.

Now I wanted to try our teletype with it. I quickly wired up our TTY to the connector and plugged in a PAC I pc board (TTY interface/resident assembler). The TTY would not work – oops, I neglected to ground the TTY select (low = select) pin on the connector. Once I did this everything worked perfectly, and I enjoyed exploring the fine operational capabilities of the unit.

Available soon from P.S.E. will be a PROM burning board, and an audio cassette interface, a CRT character generator and interface, a floppy disk interface, and BASIC (the debugged program burned into PROMs) as well as other programs in firmware. Since the Pace shares instructions with the IMP-16 (minor modification of programs might be needed), there is a lot of software already available. The *Bit Bucket*[2] newsletter is the best source of PACE and IMP-16 software. Program listings are free, source tapes $5, object tapes $3.

Overall I very much like the Pacer. I wish sockets had been provided for all the IC's and a heftier power supply had been used, however, these additions would of course increase the cost. The front panel operation and debug capabilities are the best I have seen on any commercial computer kit. I have not used any PAC II operational memory cards yet; so I can't evaluate them. However, I would highly recommend the PAC I TTY interface/resident assembler optional card. Having an assembler and dis-assembler as well as a system monitor in firmware result in relatively quick and easy assembly language, programming and debugging. The 16-bit instructions and data provide for efficient assembly language programming as well as increased accuracy. One may use words as a whole or in 8-bit bytes. Common memory and peripheral addressing result in simple quick I/O instructions.

With the Pacer's 16-bit accuracy and easy I/O and a couple of floppy disks, one could program something like Music V[3] and Score[4] for composition and playing of high fidelity music. Of course you would also need a 16-bit DAC[5]. If a very fast hardware multiply card were added as well as a fast Pace IC (rumored to be coming out from National Semi) to replace the pMOS IC, a real time FM synthesis[6] of timbre might be possible. I'm not sure if the rest of the Pacer circuits would be fast enough. Oh well – back to the 4-bit bipolar slices for real time Fourier synthesis.

PACER PRODUCT LINE RETAIL PRICE LIST (June, 1976)

Quantity	1H	2H	3H	PAC I	PAC II	PAC III	PAC IV	Fan Kit
1-3	$895	$1075	$1025	$180	$225	$50	$7	$25
4-9	$855	$1035	$ 985	$175	$245	$47	$6	$23
10-up	$820	$ 995	$ 950	$170	$235	$45	$5	$22

Club group buys would help reduce costs.

Pacer 1H – totally unassembled (not recommended by P.S.E. for beginners.
Pacer 2H – completely assembled, tested and burned in.
Pacer 3H – unassembled except for logic cards which are tested and burned in.
PAC I – TTY interface/resident assembler card.
PAC II – 2k X 16 MOS RAM card.
PAC III – prototyping card with voltage regulators.
PAC IV – dual 43 pin motherboard connector (this comes with PAC I, PAC II or PAC III).
Fan Kit – designed for general purpose use.

Footnotes for 16-bit Computer Kit article:

(1) Project Support Engineering/750 N. Mary/Sunnyvale, CA 94086.
(2) *Bit Bucket*/Compute-116/National Semiconductor/2900 Semiconductor Dr., Santa Clara, CA 95051.
(3) Described in *The Technology of Computer Music* by M. Mathews, MIT Press, Cambridge, MA 1969.
(4) "Score – A Musician's Approach to Computer Music" by L. Smith in the *Journal of the Audio Engineering Society* (JAES) Vol. 20, No. 1, Jan/Feb, '72.
(5) "Digital-to-Analog Converters: Some Problems in Producing High Fidelity Systems" by R. Talambiras, *Computer Design*, Vol. 15, No. 1, page 63, Jan, '76.
(6) J. Chowning, "The Synthesis of Complex Audio Spectra by Means of Frequency Modulation" JAES, Vol. 21, No. 7, p. 526, Sept, 1973.

A HIGH LEVEL LANGUAGE (Cont'd)

DO WHILE Statement:
Executes a group of statements a specified number of times, varying the controlling index on each repetition.

Iterative DO Statement:
Executes a group of statements a specified number of times, varying the controlling index on each repetition.

DECLARE Statement:
Defines the data types of variable (BYTE or WORD), the number of variables in arrays, initialization of variables (INITIAL), constants (DATA), strings for compile-time substitution (LITERALLY), and variables to be addressed indirectly (BASED).

Built-In Functions and Routines:
A set of built-in functions provide facilities such as rotation and shifting, decimal arithmetic time delays, binary ↔ ASCII conversion, stack operations, condition code and control flag manipulation.

Interrupt Procedures and Control:
Service routine executed when an interrupt occurs. Statements also exist for Interrupt ENABLE and DISABLE.

Assembly Language Linkage:
Assembly language procedures can be defined and linked to SM/PL programs.

Input and Output:
I/O statements that allow data to be read or written to an external device.

The charge for a copy of the manual, paper table object program and source listing is $100. To order copies of the above material please fill out the form below. Make checks payable to COMPUTE. Delivery is 6-8 weeks after your order is received.

SM/PL ORDER FORM

Name _____

Company _____

Street Address _____

City _____ State _____ Zip _____

Date _____

Enclosed is $100.00 each for _____ set(s) of SM/PL software and documentation.

UNITED STATES

COMPUTE/115
NATIONAL SEMICONDUCTOR CORP.
2900 SEMICONDUCTOR DRIVE
SANTA CLARA, CA. 95051
TEL: (408) 247-7924
TWX: 910-338-0537

EUROPE

GERMANY
National Semiconductor GmBH
808 Fuerstenfeldbruck
Industriestrasse 10
Tel: 08141/1371
Telex: 05-27649

AUSTRALIA
NS Electronics Pty Ltd
Cnr. Stud Rd. & Mtn. Highway
Bayswater, Victoria 3153
Tel: 03-729-6333
Telex: 32096

(Continued from page 10)

Further information from:
N.S.U.K. -0234-211262
Farnell Electronic, Leeds -0532-636311
Atlantic Components, Leicester -0533-65931
DTV Group Ltd., London 01-670-6166
ITT Electronic Serv., Harlow 0279-26777
Sasco Ltd., Crawley 0293-28700
Swift Hardman, Rochdale 0706-47411
Jermyn Industries, Sevendars 50144
Nsign, Reading 0734-594911

Germany
1. Date: October 13
Location: Munich
2. Date: November 8
Location: Hamburg

Further information from:
PAN Elektronik 089-6123329
EBV Elektronik 0611-720418
RTG-Distron 030-8233064
RTG-Springorum 0231-579252

SC/MP SEMINARS

Switzerland
1. Date: September 7
Location: Zurich
2. Date: September 8
Location: Geneva
Sponsor: Fenner

IMPORTANT
SEE REVERSE SIDE FOR SM/PL ORDER FORM

Holland
Rodelco is holding SC/MP 'Product Days' 1 day per week until September

Italy
1. Date: October 18
Location: Milan
2. Date: October 20
Location: Bolognia
3. Date: October 22
Location: Rome
Sponsors: Adelsy TEL (02)-4985051
Interrep TEL (06)-8124894
Interrep TEL (02)-6881783

PACE WORKSHOPS

Switzerland
Date: November 15-17
Location: Sissach
Sponsor: Fenner

France
Date: October 18-22, December 13-17
Location: Paris
Sponsor: FIME

Italy
Date: November 22-24
Location: Milan
Sponsor: C.P.M.
Via M Gioia 55
Milan
Telephone: 02-683680

For further information on these short courses, European members of Compute should contact:

Philip Huges
National Semiconductor Gmbh
808 Fuerstenfeldbruck
Industriestrasse 10
Germany
Telephone: 08141/1371
Telex: 05-27649